AFIT/GEO/ENG/90D-3

USING ERROR DETECTION AND CORRECTION CODING FOR A TURBULENT ATMOSPHERIC OPTICAL COMMUNICATIONS LINK

THESIS

Mark Allen Cloutier
Captain, USAF

AFIT/GEO/ENG/90D-3

Acknowledgments

There are many people I would like to thank for the help they have given me in preparing this thesis. My wife Heather, and children David, Kevin and Melissa were extremely supportive, in spite of the extra stress caused by Melissa being born in the winter quarter here at AFIT. Both Heather and Melissa were successful with their trying to keep my stress level bearable. Heather had a much tougher job here at WPAFB than I did. I would also like to give a special thanks to my advisor, Lt Col Dave Norman, for bringing me back whenever I got lost in the "forest," so to speak. Lastly, there are not words to describe the help that Anthony Schooler gave me, except to say that without his help this thesis would not have been finished on time. He took me from knowing nothing about UNIX, or the Sun workstations, to using batch files to process the data and prepare it for use by LaTeX. His job description says that he is a software consultant for students. I have never seen anyone do their job more effectively than he does his.

Mark Allen Cloutier

Table of Contents

List of Figures

List of Tables

AFIT/GEO/ENG/90D-3

Abstract

Probability of bit error (P_b) performance for a turbulent, atmospheric, optical communications link with convolutional error detection and correction coding was investigated. The codes investigated were all rate 1/2 codes with constraint lengths ranging from 3 to 7. It was assumed that the coded data stream was sufficiently interleaved so that the turbulent channel could be considered memoryless. A theoretical bound on P_b, given by Viterbi for a memoryless channel, was used to plot curves of P_b versus the signal intensity of the laser transmitter. These curves were plotted for three different levels of atmospheric turbulence and three different levels of background light.

USING ERROR DETECTION AND CORRECTION CODING FOR A TURBULENT ATMOSPHERIC OPTICAL COMMUNICATIONS LINK

I. Introduction

1.1 Background

John Holt made it clear in the executive summary for the HAVE LACE test program that atmospheric optical communications hold great promise for the Air Force.

> Laser communications offers many advantages over radio frequency (RF) systems for airborne applications. Most of these advantages stem from the frequency of the carrier, which is four orders of magnitude greater than any other being considered for airborne use The higher frequency allows reduction in antenna size and/or improvements in antenna gain. High gain antennas permit these systems to provide long range, high data rate links using low power solid state transmitters. Another advantage of the optical frequency is the ease of eliminating the backlobes and reducing the sidelobes by orders of magnitude over RF systems. This makes laser communications inherently jam resistant and difficult to intercept (21).

Large data rates, low probability of intercept (LPI), and frequency availability are benefits that have driven the fiber optics industry through the roof, but only recently has a lot of work been done to overcome, or least work around the problems that atmospheric effects cause. The two major atmospheric effects are the extinction of the optical beam due to scattering and absorption, and the disruption of the beam due to clear-air turbulence (19). The Air Force test program HAVE LACE

was specifically aimed at investigating atmospheric effects on high altitude, air-to-air optical communications. Robert Feldman used the data from HAVE LACE to investigate turbulence induced scintillation of the optical beam, and to verify a theoretical model for describing the clear-air turbulence (13). John Holt investigated the effects of scattering and absorption from measurements taken in the HAVE LACE test flights (20). The Air Force Geophysics Lab has also done a lot of work in the characterization of atmospheric turbulence (2, 4, 17, 46). However, most of their experimental work to date, has concentrated on the ground-to-space vertical path. The only work they have done on horizontal paths was at low altitudes (2, 34, 46). They are just now starting to study the use of a laser radar for remote measurement of turbulence parameters for a high altitude, horizontal path (3).

In order to predict a communications systems's performance, the communications engineer must be able to model the channel. For a digital laser communications system, predicting performance means being able to predict the probability of a bit, or symbol error at the receiver. Modeling the channel means being able to mathematically characterize atmospheric turbulence effects. If the effects of the channel on the signal can be characterized, various techniques are available to combat these effects.

Feldman suggested in his thesis that error detection and correction (EDAC) coding could be used for improving performance (13). The idea for this thesis was more concisely defined through discussions with Mr. Feldman, who was representing the Wright Research and Development Center (WRDC) as the sponsor for this thesis. A typical communications system is designed to provide a certain minimum performance, maximum probability of error in the digital case, under the worst channel conditions anticipated. Improvement in the channel means a decreased probability of error at the same range, or the same probability of error at an increased range. In the LPI environment, extended range or improved performance is not necessarily a benefit. The minimum probability of error required for reliable decoding of the signal

should occur at the receiver, and no further. Although adaptive coding by itself does not provide a LPI capability, when combined with adaptive power control, it would allow the transmitter to be operated at lower power levels which would provide a LPI capability. WRDC wants to investigate the use of adaptive EDAC to combat changing turbulence in the atmospheric channel. If the turbulence increases, thus raising the probability of error, then the EDAC will be increased to provide more coding gain. Conversely, if the turbulence decreases, the coding gain can be decreased so as to prevent the signal from being decoded beyond the intended receiver. A secondary benefit of adaptive EDAC is the maximization of data throughput, since the amount of coding redundancy in the data stream is minimized.

1.2 Problem Statement

There is no documented study of the use of adaptive EDAC in an atmospheric optical communications link for the purpose of compensating for the time varying fading caused by turbulence.

1.3 Objective

The objective of this thesis is to show that EDAC can adaptively compensate for the time varying fading that occurs in the clear-air turbulent atmospheric channel, allowing the communications engineer to design the link so that the signal is detectable at the receiver, but not beyond the receiver.

1.4 Summary of Current Knowledge

Extensive research has been done in the areas of atmospheric turbulence (1, 2, 4, 17, 23, 34, 41, 44), and optical communications in space, where the channel is considered memoryless. In a memoryless channel, errors occur randomly with complete statistical independence between them (15, 36). The atmospheric channel, however, is not a memoryless channel. Errors occur in bursts and are not statistically inde-

pendent. This makes the theoretical analysis of the probability of error very difficult (12:993). As a result, documented research into the use of EDAC for the turbulent atmospheric channel is sparse. Most of it is experimental in nature, comparing actual measured system performance to the theoretical bounds on the probability of error. These bounds are derived for a memoryless channel (5, 12, 25). Cheng's dissertation (6) was the only source found that did a theoretical performance analysis of an optical communications system over a turbulent, atmospheric channel. He did not analyze the probability of error, however. His measure of performance was the cutoff rate for the channel, which he was able to theoretically determine.

1.5 Scope

This thesis will describe a communications system that uses a direct detection receiver. Consequently, the turbulence effect of interest in Chapter II is scintillation since it affects the intensity of the received beam. Phase distortions will not need to be considered. As for the EDAC, the coding gains will be analyzed for convolutional codes of varying constraint lengths, using the Viterbi algorithm (47:235-239) for soft decision decoding. The constraint length of the convolutional codes will be limited to 10 since present technology can not provide real-time decoding with the Viterbi algorithim for anything longer (40:338).

1.6 Methodology

Due to the "prohibitively complex" (12:993) mathematics needed to theoretically determine the probability of bit error for the clear-air turbulent optical channel, there are two options available for determining the probability of bit error curves. Experimental measurements can be taken, or, it can be assumed that the information signal has been sufficiently interleaved so as to make the channel appear memoryless. Equipment and time limitations mandate the second option for this thesis. Davidson's results (12) will be used to determine what can be considered sufficient

interleaving. Davidson showed that with soft decision Viterbi decoding and low atmospheric turbulence, an interleaving delay of about 5 milliseconds was sufficient to bring the measured probability of error curves very close to the curves predicted for a memoryless channel (12:1001). Assuming a memoryless channel, theoretical values for various convolutional codes will be calculated and plotted. Then, assuming sufficient interleaving, these plots can be used to compare the probability of bit error for the different coding schemes.

1.7 Materials and Equipment

The simplifications resulting from the assumptions described in section 1.6 will allow the calculations to be made on available AFIT computer resources, using available software. A mathematical analysis package such as MathCad, Math Station, or Mathematica will be needed, as well as a matrix oriented analysis tool such as MATRIXx or Matlab. All of these are available on AFIT's Sun workstations except for MathCad which is available on the AFIT PC network server.

1.8 Organization of the Thesis

Chapters II and III review the theory needed for examining a digital laser communications system that uses EDAC to combat scintillation effects. Chapter II covers the atmospheric effects, concentrating on turbulence induced scintillation, while Chapter III covers the principles of EDAC, concentrating on convolutional coding. Chapter IV gives a detailed description of the methodology used to derive the equations needed for calculating the theoretical probability of bit error curves. Chapter V presents the results of the calculations, while Chapter VI contains the conclusions and recommendations for possible follow on research efforts.

II. Turbulence Theory

2.1 Introduction

The atmosphere has two basic effects on light propagation, extinction and disruption (19). In the case of laser communications, extinction means that energy from the laser transmitter is prevented from reaching the receiver or detector, while disruption means that the energy that reaches the detector is somehow distorted from the form it had when it left the transmitter. Two ways that the atmosphere can prevent the laser energy from reaching the detector are absorption and scattering. Water, carbon dioxide and other molecules in the atmosphere will absorb light, although the amount of absorption is wavelength dependent. Figure 2.1 shows that

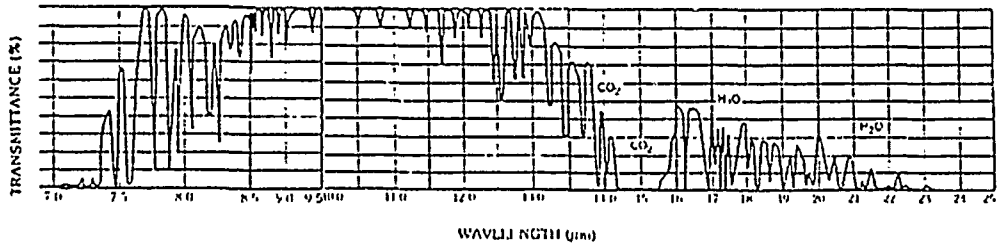


Figure 2.1. Absorbtion Spectrum

there are certain wavelength regions where absorption is minimized. These regions are known as absorption windows. The proper selection of wavelength will allow the system to operate in one of these windows. Particles in the atmosphere, called aerosols, will cause scattering of the laser energy, preventing any measureable amount of energy from reaching the detector. If absorption and scattering can be considered negligible, then the atmosphere is described as being "clear air turbulent" (CAT) (18). Goodman does a good job of distinguishing between the CAT atmosphere, and an atmosphere where absorbtion and scattering are not negligible.

> We shall always assume that the scale sizes are much larger than the wavelength of the radiation being used. This assumption eliminates from consideration problems involving imaging through clouds or aerosols, for which the scale sizes of the inhomogeneities are comparable with or smaller than an optical wavelength and for which the refractive index changes are sharp and abrupt. This latter class of problems may be referred to as "imaging through turbid media" whereas we are concerned here with "imaging through turbulent media" for which the refractive index changes are smoother and coarser (18:362).

The "sharp and abrupt" changes in the index of refraction cause large changes in the direction of the radiation, which is nothing more than scattering. Goodman also makes the assumption that absorption is negligible in his analysis of the CAT atmosphere (18:385).

2.2 Turbulence Effects on Laser Beam Propagation

Knowing what conditions are necessary to define a CAT atmosphere, what is the effect of such an atmosphere on the propagation of the laser beam? The two primary effects are scintillation and coherence degradation. Pratt defines scintillation as "small scale destructive interference within the beam cross-section, causing variations in the spatial power density at the receiver (37:133)." This same interference also causes a coherence degradation since the interference is happening randomly. However, in this thesis, the use of a direct detection receiver is assumed which is usually the case in present laser communications links. Since only intensity variations affect the probability of bit error with this type of receiver, coherence, or phase degradation of the beam is not a problem.

2.3 Modeling of Scintillation

Propagation of light in the atmosphere, just as everywhere else, can be described by Maxwell's equations. Assuming that the atmosphere has zero conductivity and unit magnetic permeability, and that the electric field is sinusoidal with a

time dependence of $e^{i\omega t}$, Maxwell's equations reduce to

$$\nabla \cdot H = 0 \tag{2.1}$$

$$\nabla \times E = ikH \tag{2.2}$$

$$\nabla \times H = -ikn^2 E \tag{2.3}$$

$$\nabla \cdot (n^2 E) = 0 \tag{2.4}$$

where E and H are the electric and magnetic fields respectively, n is the index of refraction, k is the wave number, ω/c, where ω is the radian frequency and c is the speed of light, and i is $\sqrt{-1}$. Using vector calculus, these equations can be reduced to the more familiar wave equation:

$$\nabla^2 \vec{E} + k^2 n^2 \vec{E} + 2\nabla(\vec{E} \cdot \nabla log n) = 0 \tag{2.5}$$

The last term on the left hand side describes the depolarization effects. This term can be considered negligible for wavelengths (λ) $\leq l_o$ where l_o is the inner scale of the turbulence (10, 27, 38, 42, 43, 45). For atmospheric turbulence, $l_o \geq$ 1mm, so with optical waves where λ is on the order of a micron, this is a valid assumption. If you also consider the components of the electric field separately, equation 2.5 can be written in its scalar form as:

$$\nabla^2 E + k^2 n^2 E = 0 \tag{2.6}$$

If the channel was the space channel, where $n \approx 1$, or in a fiber with $n \approx 1.55$, then solving this equation is fairly straightforward. Unfortunately, the atmosphere, due to turbulence, is inhomogeneous and anisotropic. As a result n is neither constant, nor even deterministic. As Clifford states:

> The refractive index fluctuations that we are considering are those that

> result from the naturally occurring random fluctuations in wind velocity called turbulence. Because turbulence is a random process, it must be described in terms of statistical properties. It is hopeless to attempt to predict the velocity of a given parcel of air as it is buffeted about in the open atmosphere. In fact, it is nearly impossible to do a complete statistical analysis. Therefore, we settle for a more meager description. (9:11)

Obviously the first task is to statistically characterize n. The usual way of describing a random process is in terms of its mean and autocorrelation functions (44:3). Assuming wide sense stationarity, the Fourier transform of the autocorrelation function can then be used to describe the power spectral density of the process. A good example of this is the characterization of white noise. If white noise is sampled at any two time instants, there is complete statistical independence between the samples. Therefore, the autocorrelation is zero. Only when the time difference between samples, τ, is zero, is the autocorrelation non-zero. The autocorrelation function is then given by a delta function at $\tau = 0$ with a magnitude that is related to the strength of the fluctuations in the noise, given by the variance of the gaussian noise, as shown in Figure 2.2. The Fourier transform of this delta function is a flat,

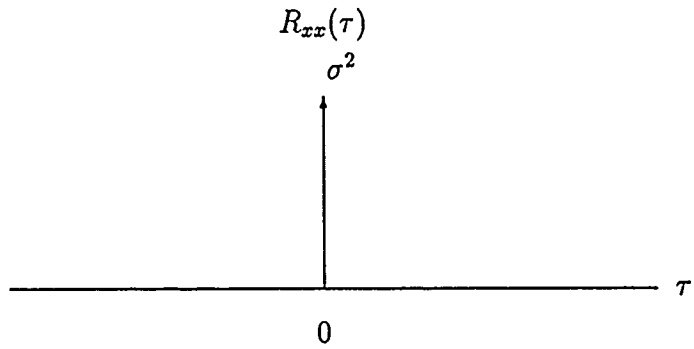


Figure 2.2. Autocorrelation Function for White Noise

infinitely wide power spectral density whose magnitude is inversely proportional to the strength of the fluctuations in the noise. The power spectral density is shown in Figure 2.3. As was mentioned above, one condition for being able to take the

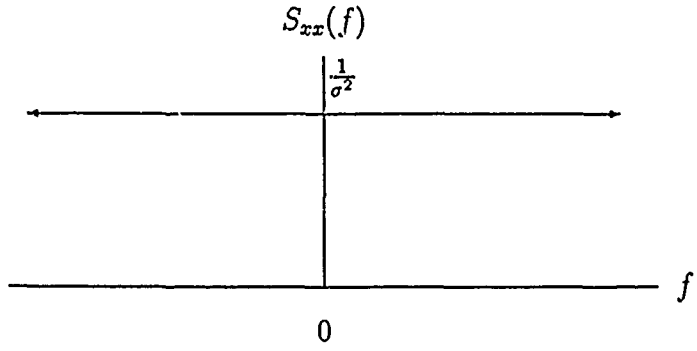


Figure 2.3. Noise Power Spectral Density

Fourier transform is that the random process be wide sense stationary. That is, its mean value must be independent of time, and its autocorrelation must be a function only of the time difference, $\tau = t_1 - t_2$, not the actual times t_1 and t_2.

Just as the power spectral density of white noise is used in analyzing RF communications links, a power spectral density for the fluctuations in n is needed in order to analyze the atmospheric optical communications link. Unfortunately, most real random processes are not stationary, so as in all applications of theory to the real world, the question is, how good is the approximation? Herein lies the problem in characterizing the random fluctuations in n. It is not even close to being a stationary random process.

> Atmospheric turbulence always contains large scale components which usually destroy the homogeneity and isotropy of fields of the meterological variables; moreover, these components cause the meteorological fields to be non-stationary. (44:19)

In general, the random fluctuations of n are a function of position, wavelength, and time: $n = n(\vec{r}, \lambda, t)$. However, the wavelength dependence is usually negligible in the atmosphere. The time dependence is also negligible since "The time required for light to propagate through the atmosphere is only a fraction of the fluctuation

time of the random refractive index component." (18:385-6) All that is left is the spatial variation of n, $n(\vec{r})$. The autocorrelation function of n can be written as:

$$B_n(\vec{r}) = E\{n(\vec{r_1})n(\vec{r_2})\} \tag{2.7}$$

where $\vec{r} = \vec{r_1} - \vec{r_2}$. If n was a stationary process, then the power spectral density of the fluctuations could be obtained by Fourier transforming equation 2.7. Unfortunately, as mentioned earlier, n is not a stationary process. The mean value changes with $\vec{r}$, and, due to the inhomogeneities of the atmosphere, the autocorrelation function depends not only on the vector distance between sample points, $\vec{r_1} - \vec{r_2}$, but on the location of the sample points, $\vec{r_1}$ and $\vec{r_2}$, as well.

Kolmogorov's solution to this problem was the structure function. The structure function for n is the square of the average difference in n at the two points, rather than the average of the product of n at the two points as is the autocorrelation function. Tatarski describes the benefit of the structure function as:

> The difference between the values of the $f(\vec{r})$ at two points $\vec{r_1}$ and $\vec{r_2}$ is chiefly affected only by inhomogeneities of the field f with dimensions which do not exceed the distance $|\vec{r_1} - \vec{r_2}|$. If this distance is not too large, the largest inhomogeneities have no effect on $f(\vec{r_1}) - f(\vec{r_2})$ and therefore the structure function
>
> $$D_f(\vec{r_1}, \vec{r_2}) = \overline{\left[f(\vec{r_1}) - f(\vec{r_2})\right]^2}$$
>
> can depend only on $\vec{r_1} - \vec{r_2}$. At the same time the value of the autocorrelation function is affected by inhomogeneities of all scales, so that for the same values of $\vec{r_1}$ and $\vec{r_2}$ the correlation function can depend on each of the arguments separately and not just on the difference $\vec{r_1} - \vec{r_2}$. (44:19)

The structure function takes advantage of a process which has stationary increments. This is true for n since the atmosphere can be considered locally homogeneous within the distance $|\vec{r_1} - \vec{r_2}|$. In describing turbulence, whether it is atmospheric, oceanic or some other medium, the parameter whose fluctuations are most apparent is velocity.

In an airplane, the velocity changes are felt as buffeting. In the water, currents and eddies display the fluctuating nature of the velocity. Other parameters that depend directly or indirectly on the velocity changes, can be understood better by understanding the velocity fluctuations. Kolmogorov showed that as long as $|\vec{r_1} - \vec{r_2}|$ is small enough, the structure function of the velocity is given by

$$D_{rr} = C_v^2 r^{2/3} \tag{2.8}$$

where C_v^2 is defined as the structure constant of the velocity. The structure constant is a measure of the strength of the fluctuations (9:12), just as the variance of gaussian noise is a measure of the strength of the fluctuations in the noise. Equation 2.8 is similar to Figure 2.2 in that D_{rr} is proportional to C_v^2, while $R_{xx}(\tau)$ is proportional to σ^2.

Unfortunately there is no direct relationship between the velocity fluctuations and the fluctuations in n. The index of refraction depends on pressure, temperature, and wavelength as shown in equation 2.9 (9:10).

$$n - 1 = 77.6(1 + 7.52 \cdot 10^{-3}\lambda^{-2})(P/T) \cdot 10^{-6} \tag{2.9}$$

where P is given in millibars, T is in degrees Kelvin and λ is in microns. The left hand side of the equation is written in a form which directly gives the value of the fluctuation of n as it varies about its mean free space value of 1. Equation 2.9 is only good for over land paths since it neglects the effects of humidity. Over the water, a pressure humidity product factor makes an appreciable contribution (9).

Corrsin showed that potential temperature obeys the same two-thirds law as the velocity fluctuations (11). How then can n be related to potential temperature? Clifford describes the quantity potential temperature as: "...the difference between the absolute temperature T and the changes in temperature with height caused by an adiabatic lapse rate ... $\Theta = T - \gamma_a z$" (9:12). Taking the partial derivative of

equation 2.9 with respect to pressure and temperature and evaluating at $\lambda \cong .6\mu m$, which corresponds to red light (9:13) we get:

$$\delta n = 79P/T\,(\delta P/P - \delta T/T) \cdot 10^{-6} \tag{2.10}$$

If it is also assumed that the pressure variations are negligible for a fixed altitude then the fluctuations in n, at a given height, are a function only of the fluctuations in temperature at that height, and the assumed constant pressure at that height. At a fixed altitude, the fluctuations in actual temperature also obey the $\frac{2}{3}$ law. Therefore δn does also (9:13). Consequently, the structure function for n can be written as:

$$D_n(r) = C_n^2 r^{2/3} \tag{2.11}$$

where C_n^2 is given by (9:13):

$$C_n^2 = \left(\frac{79P}{T^2} \times 10^{-6}\right)^2 C_T^2 \tag{2.12}$$

Although the fluctuations of n itself are not a stationary random process, the fluctuations of n about its mean value, which from here on will be designated as n', is a stationary process if the atmosphere can be considered locally homogeneous within the same constraints on $|\vec{r_1} - \vec{r_2}|$ that applied to the structure function. The power spectral density of the fluctuations can now be found by Fourier transforming the autocorrelation function of n' (9:18).

$$\Phi_{n'}(\kappa) = \frac{1}{(2\pi)^3} \int B_{n'}(r) e^{-i\kappa \cdot r} d^3 r \tag{2.13}$$

Assuming that the atmosphere can be considered locally isotropic, Equation 2.13 can be rewritten in scalar form as :

$$\Phi_{n'}(\kappa) = \frac{1}{2\pi^2 \kappa} \int_0^\infty r B_{n'}(r) sin(\kappa r) dr \tag{2.14}$$

Again using the assumption that the atmosphere is locally homogeneous and isotropic, equation 2.14 can be written in terms of the structure function $D_{n'}(r)$, since under these conditions:

$$B_{n'}(0) - B_{n'}(r) = \frac{1}{2} D_{n'}(r) \tag{2.15}$$

Combining equations 2.14, and 2.15, $D_{n'}(r)$ can be written in terms of $\Phi_{n'}(\kappa)$ as:

$$D_{n'}(r) = 8\pi \int_0^\infty \kappa^2 \Phi_{n'}(\kappa) \left[1 - \frac{sin(\kappa r)}{\kappa r}\right] d\kappa \tag{2.16}$$

Inverting equation 2.16 is not as trivial as Fourier transforming a delta function, but it can be done with the judicious use of derivatives (44) giving:

$$\Phi_{n'}(\kappa) = \frac{1}{4\pi^2\kappa^2} \int_0^\infty \frac{sin(\kappa r)}{\kappa r} \frac{d}{dr} dr \left[r^2 \frac{d}{dr} D_{n'}(r)\right] dr \tag{2.17}$$

Substituting equation 2.11 into equation 2.17, and integrating, $\Phi_{n'}(\kappa)$ can be written in terms of $C_{n'}^2$:

$$\Phi_{n'}(\kappa) = .033 C_{n'}^2 \kappa^{-11/3} \tag{2.18}$$

Equation 2.18 is governed by the conditions of local homogeneity and isotropy. The resulting limits on κ are given by Kolmogorov's inertial subrange model, $2\pi/L_o \ll \kappa \ll 2\pi/l_o$, where l_o is the inner scale of the turbulence and L_o is the outer scale of the turbulence. Clifford states that : "Near the ground l_o is on the order of millimeters and L_o of the order of meters. Both quantities appear to increase with height above the ground (9:12)." Equation 2.18 needed some corrections in order to accurately model the physical reality of the power spectral density of the fluctuations in n'. Goodman descibes these corrections as: (18:389)

> When κ reaches ... κ_m, the form of $\Phi_{n'}(\kappa)$ again changes. Turbulent eddies smaller than a certain scale size dissipate their energy as a result of viscous forces, resulting in a rapid drop in $\Phi_{n'}$ for $\kappa > \kappa_m$. The scale size $l_o \cong 2\pi/\kappa_m$ is referred to as the inner scale of the turbulence. A typical value for l_o near the ground is a few millimeters. Tatarski

includes the rapid decay of $\Phi_{n'}(\kappa)$ for $\kappa > \kappa_m$ by use of the model

$$\Phi_{n'}(\kappa) = 0.033 C_n^2 \kappa^{-11/3} e^{\frac{-\kappa^2}{\kappa_m^2}}. \tag{2.19}$$

This equation is a reasonable approximation provided κ_m is chosen to equal $5.92/l_o$ and $\kappa > \kappa_o$.

Spectra 2.18 and 2.19 both have nonintegrable poles at the origin. In fact, since there is a finite amount of air associated with the Earth's atmosphere, the spectrum can not become arbitrarily large as $\kappa \rightarrow 0$. To overcome this defect in the model, a form known as the von Karman spectrum is often adopted. The spectrum is then expressed as

$$\Phi_{n'}(\kappa) = \frac{0.033 C_{n'}^2}{(\kappa^2 + \kappa_o^2)^{11/6}} e^{\frac{\kappa^2}{\kappa_m^2}} \tag{2.20}$$

A plot of equation 2.20, showing the inertial subrange limits is given in Figure 2.4. $\Phi_{n'}$, normalized by $C_{n'}^2$, is plotted against the spatial frequency, κ, which is inversely proportional to a spatial distance.

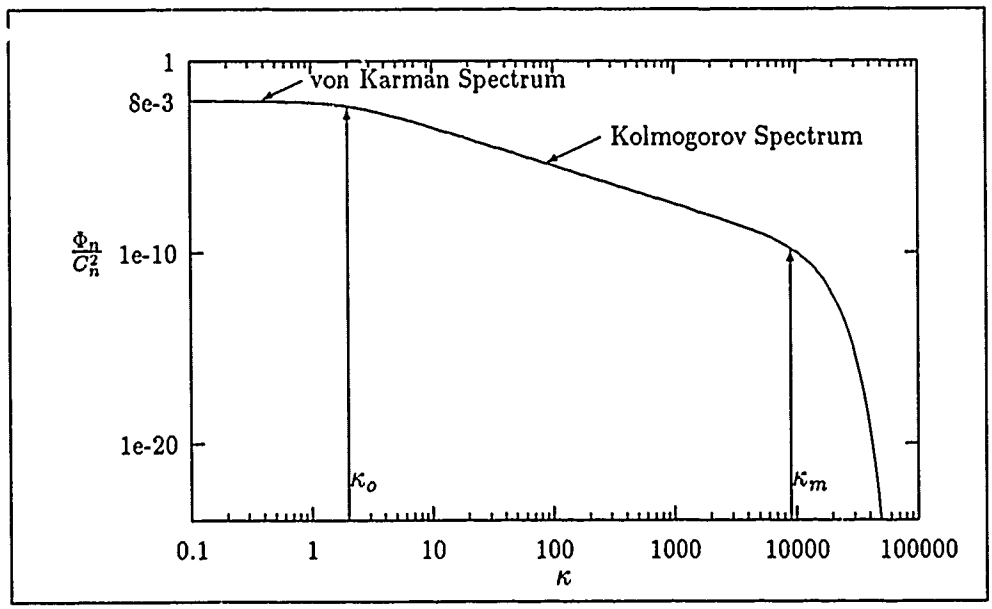


Figure 2.4. Power Spectral Density of Fluctuations in n
(18:390)

Clifford then goes through the solution of equation 2.6 using the method of small pertubations and Rytov's approach of taking the log of the amplitude and representing it as χ (9:26-34). The resulting equations for the power spectral densities of the log-normal amplitude, F_χ and phase spectra, F_s are given in equation 2.21.

$$\begin{bmatrix} F_\chi(\kappa,0) \\ F_s(\kappa,0) \end{bmatrix} = \pi k^2 L \left[1 \mp \left(\frac{k}{\kappa^2 L}\right) sin\left(\frac{\kappa^2 L}{k}\right)\right] \Phi_n(\kappa) \tag{2.21}$$

The minus sign gives the two dimensional power spectral density of the log-normal amplitude spectra and the plus sign gives the two dimensional phase spectra. The justifications for Rytov's transformation is simply that the resulting log-normal distribution of the amplitudes matches the empirical data for low levels of turbulence. As the level of turbulence increases, either because of increasing C_n^2, or longer path length, this model breaks down as the received signal can no longer be described by a log-normal distribution. More research has been done to extend the model (42:45-106), but that is beyond the scope of this thesis.

III. Error Detection and Correction Coding

3.1 Introduction

EDAC got its real start with Shannon's theorem on channel capacity in 1948. Shannon showed that for a data rate that was less than the capacity of the channel, an arbitrarily small probability of bit error could be acheived through the use of EDAC.

> Let a discrete channel have the capacity C and a discrete source the entropy per second H. If $H \leq C$ there exists a coding system such that the output of the source can be transmitted over the channel with an arbitrarily small frequency of errors (39:39).

Shannon's theorem said nothing about how to do the coding, just that it could be done. The development of the codes was left to others. There are several good textbooks on the principles of EDAC (8, 28, 40, 47). This chapter will provide a summary of the theory needed for understanding the use of EDAC in a clear-air turbulent, atmospheric optical communications link.

3.2 Block Codes vs. Convolutional Codes

There are two basic types of codes that are used in existing communications systems, block codes and convolutional codes. According to Viterbi, convolutional codes "invariably outperform block codes of the same order of complexity" (49). Convolutional codes are also easier to decode with a soft decision decoder than block codes (40:231). Soft decision decoding is simply a means of obtaining more information from the output of the demodulator. Hard decision decoding puts out a one or a zero based on the relationship between the output level and the threshhold level. If the output is above the threshhold, a one is output. If the output is below the threshhold a zero is output. No information is obtained about how "good" the

decision is. Soft decision decoding quantizes the output. Instead of only having one threshhold, there are $2^k - 1$ threshholds, where k is the number of quantization bits. The output can now contain information on how close the output is to a zero or a one. Figure 3.1 demonstrates this for $k = 3$ (40:331). Eight level quantization

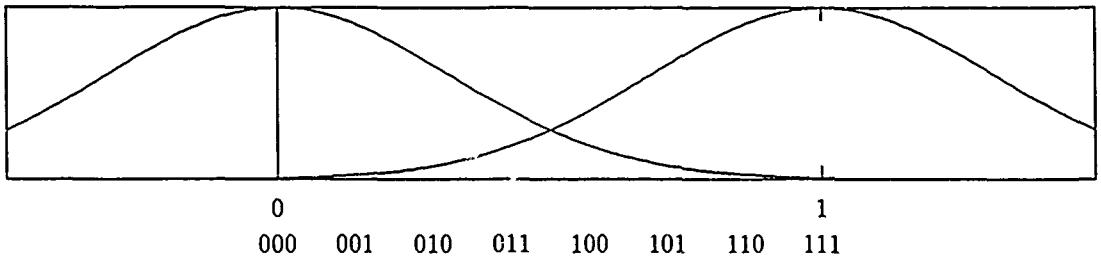


Figure 3.1. Soft Decision Decoding

($k = 3$) is usually used since it gives a 2 dB improvement in performance, whereas infinite quantization only provides another .2 dB of improvement (40:331). Soft decision decoding has been shown to provide between 1 and 2 dB more coding gain over hard decision decoding, when used on a turbulent atmospheric channel (12). These factors have driven the use of convolutional codes, with soft decision decoding, for use on an atmospheric optical communications link.

3.3 Convolutional Code Description Methods

The usual implementation of a convolutional encoder is a sequence of shift registers with modulo-2 adders connected to certain registers in the sequence. A classic example used in several textbooks (8, 40, 47), was first used by Viterbi (49). It is shown in Figure 3.2. The code, or encoder, is described by representing the taps octally. This code for example, would be represented by (5,7), since the top modulo-2 adder has taps coming from the third bit, (numbered from right to left) and from the first bit. In binary this would be 101, octally a 5.

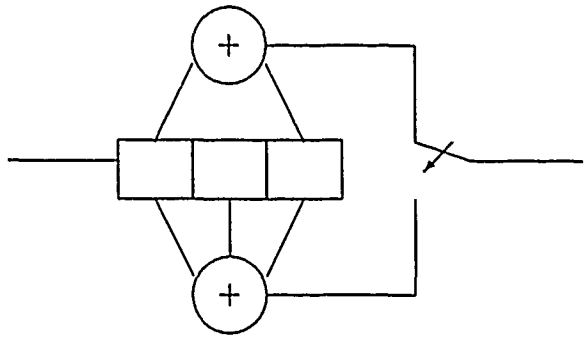


Figure 3.2. K = 3 Encoder Example

The code can also be represented by a tree, as shown in Figure 3.3. Usually the encoder is assumed to start with all the shift registers containing zeroes. An input bit of zero is represented by the upper branch while a one follows the lower branch. The output codewords are shown on each branch. As an example, an input message of 10101 would result in a transmitted message of 0001101011 where the rightmost bit of the input message is the first bit into the encoder and the rightmost bit of the transmitted message is the first bit out. The path for this message is shown in Figure 3.3. The tree diagram becomes unwieldy for an input sequence of any appreciable length, since the number of branches increases as 2^L, where L is the length of the input message. Fortunately, however, the tree starts to repeat itself after K branchings, where K is the number of shift registers in the encoder, known as the constraint length of the code. As can be seen in Figure 3.3, the fourth branching duplicates the eight codewords from the upper and lower halves of the tree. This means that the output codeword doesn't depend on the K+1 previous bit. Thus the constraint length determines the memory of the encoder. This feature allows the tree diagram to be simplified to a trellis diagram as shown in Figure 3.4. The trellis diagram shows the possible state transitions, the resulting codeword, and the input message bit that caused that state transition. An input bit of 0 is represented by a solid line, and an input bit of 1 is reprented by a dashed line. As an example, again assuming the encoder is started with all zeroes, the two possible transitions

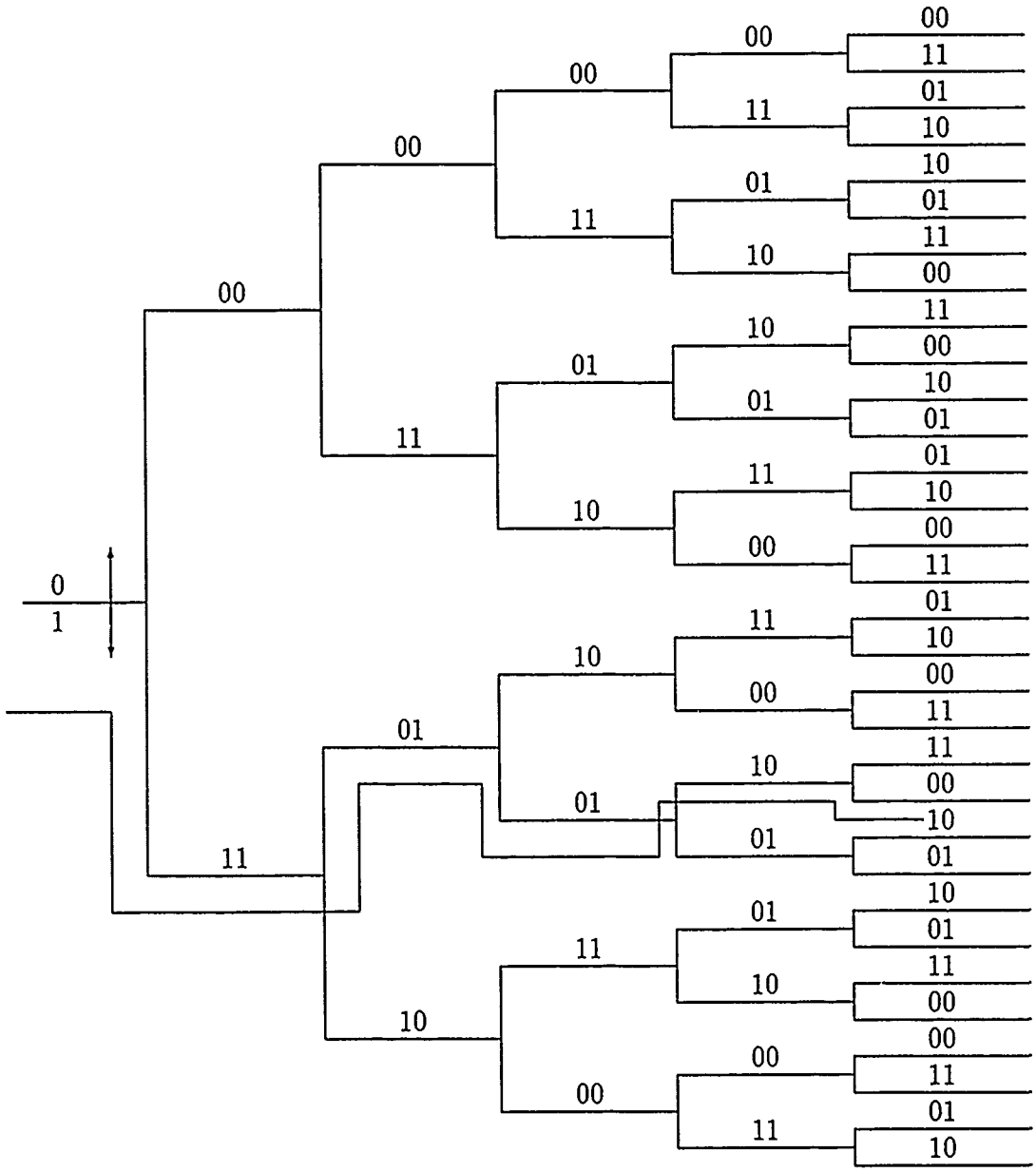


Figure 3.3. Tree Diagram for Example Encoder

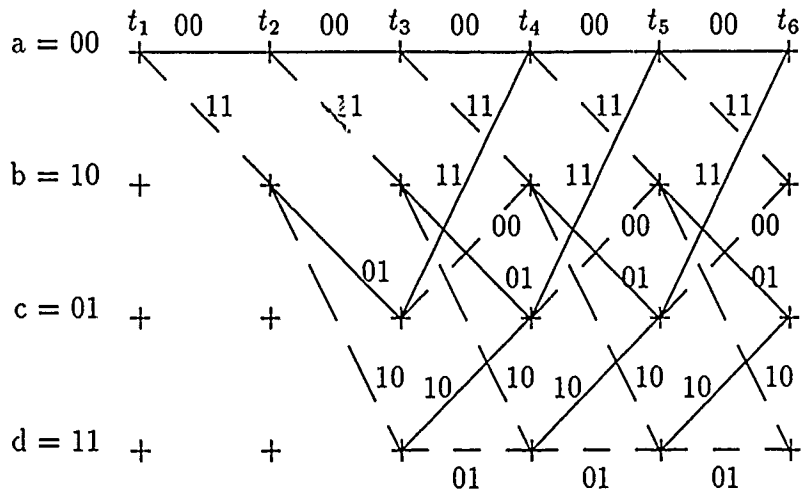


Figure 3.4. Trellis for Example Encoder

are to state 10, or to remain in state 00. The transition to 10, caused by an input bit of one, results in an output codeword of 11, whereas an input bit of 0 causes the encoder to remain at the 00 state, resulting in an output codeword of 00. It will be discussed later how useful the trellis diagram is in visualizing the decoding of a convolutional code, but for now a clearer picture of the state transitions is given by a state diagram. The state diagram for the encoder of Figure 3.2 is given in Figure 3.5.

The state diagram is the most compact way of representing the encoder since its size depends only on the number of states and not on the length of the input message as the tree and trellis diagrams do. The number of states is given by 2^{K-1} so it is still going to be complicated for $K \geq 4$. A solid line still represents an input bit of 0, and the dashed line still represents a 1, as in the trellis diagram. All possible state transitions are shown with their respective output codewords. The diagrams shown in figures 3.3, 3.4, and 3.5 are patterned after those given by Sklar (40:322–326). The only difference is, the taps on Sklar's encoder are given octally by (7,5) instead of (5,7), causing the order of the bits in the output codeword to be swapped. The codewords 00 and 11 are thus unchanged, while the 01 becomes a 10, and vice versa.

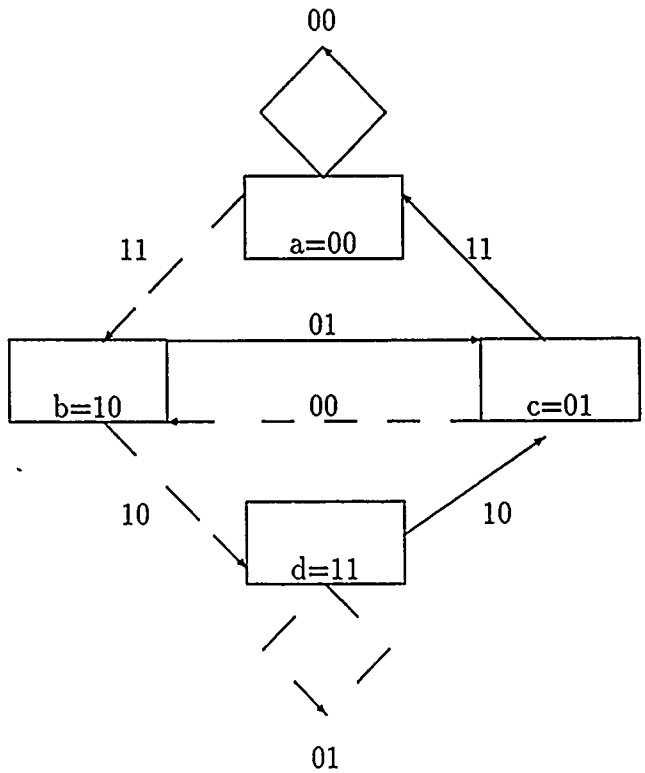
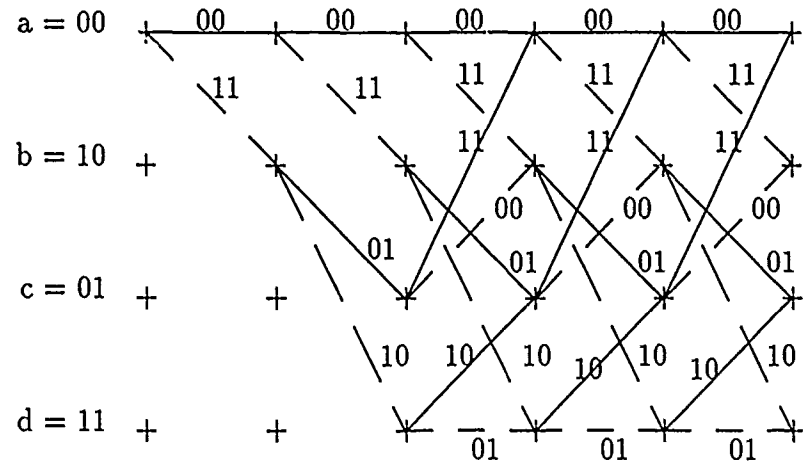


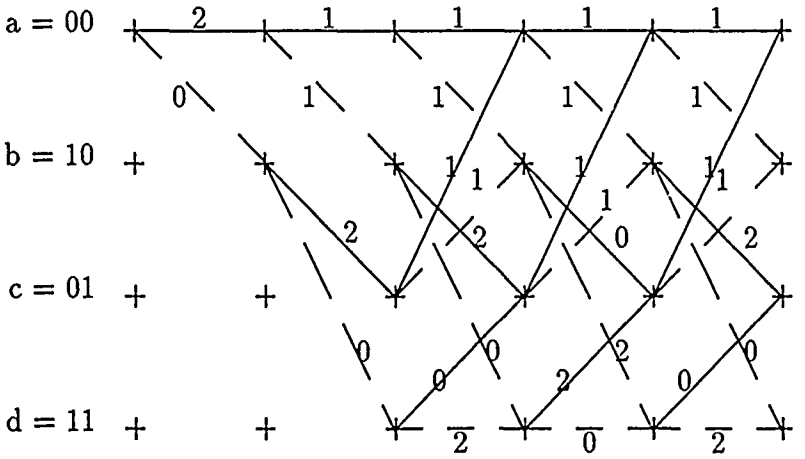
Figure 3.5. State Diagram for Example Encoder

As mentioned earlier, the trellis is a useful diagram for visualizing the decoding process. This is demonstrated in Figure 3.6 which shows both the encoder and decoder. The branches of the encoder trellis diagram are still labeled with the output codeword resulting from that state transition path. The decoder trellis branches, however are labeled with an appropriate path metric describing the conditional probability of that path being the correct path given the input codeword. The basis of Viterbi decoding can be summed up in one sentence. "If any two paths in the trellis merge to a single state, one of them can always be eliminated in the search for the optimum path (40:334)." The elimination is done by comparing the cumulative path metric for the two paths. The path with the higher metric will be discarded. If the path metrics are equal, one of them is discarded by an arbitrary rule. The Viterbi algorithm has been shown to be an optimum decoding scheme (35). This procedure can be seen at time t_4 of the decoder trellis in Figure 3.6. In this example the Hamming distance, defined as the difference in the number of ones between the received



Encoder Trellis

Input data :	1	1	0	1	1
Output:	11	10	10	00	10
Received :	11	10	10	01	10



Decoder Trellis

Figure 3.6. Encoder and Decoder Trellis Diagram

codeword and the transmitted codeword, is used as the path metric. Typically in a optimal decoder the metric that is used is some type of maximum likelihood metric, but for simplicity's sake, this example uses the Hamming metric. There are two paths merging at state a, at time t_4. The first path, which stayed in state a for three time periods, has a cumulative path metric of 4. The other path which goes through states b and c before returning to state a, has a cumulative path metric of 3. The first path has a higher path metric, and therefore can be eliminated. As the decoder progresses deeper into the trellis, more paths are eliminated until only one path remains from time t_1 to time t_2, and the first bit can then be output. Figure 3.7 shows the surviving paths after three and four codewords have been received. The cumulative path metrics for the upper and lower paths arriving at a node are shown to the right of the final node. For example, at time t_5 the node for state a has an upper path whose cumulative path metric is 4 and a lower path whose cumulative path metric is 1. Figure 3.8 shows the surviving paths at times t_6 and t_7. Since there are still two paths going from time t_2 to time t_3, no output bit can be decided yet. Figure 3.9 shows the surviving paths at time t_8. The codeword received at time t_8 allows the elimination of all but one path between time t_3 and t_4, allowing the output of a zero for the third message bit. Figure 3.10 shows that no more output bits can be decided upon, since there are still two paths between time t_4 and t_5. Figure 3.11 demonstrates that there still is not enough progression into the trellis to decide on another output bit. Figure 3.12 shows that more progression is still needed. Figure 3.13 shows that after picking the paths with the higher cumulative path metrics at time t_{11}, only one path remains for the five bit durations that correspond to the five message bits. It also shows that the error in the fourth codeword was corrected.

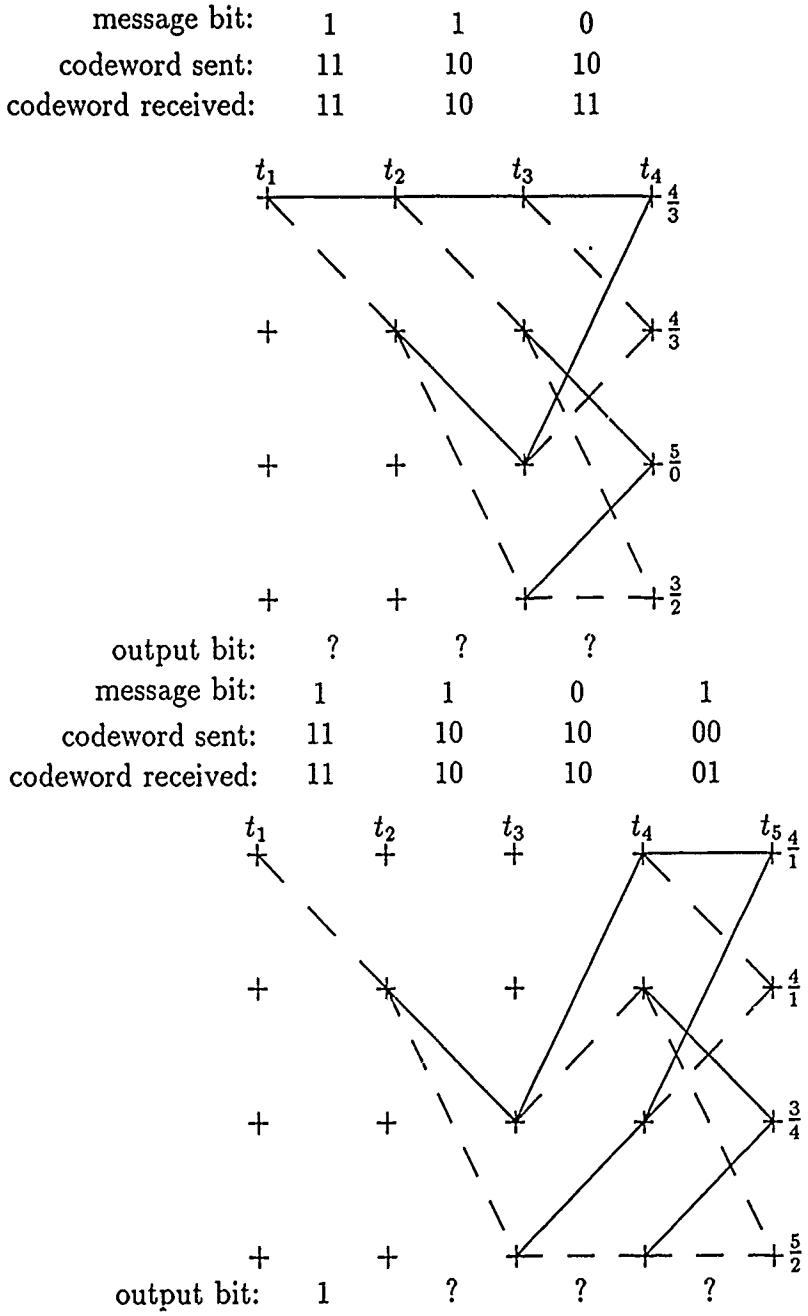


Figure 3.7. Surviving Paths at t_4,t_5

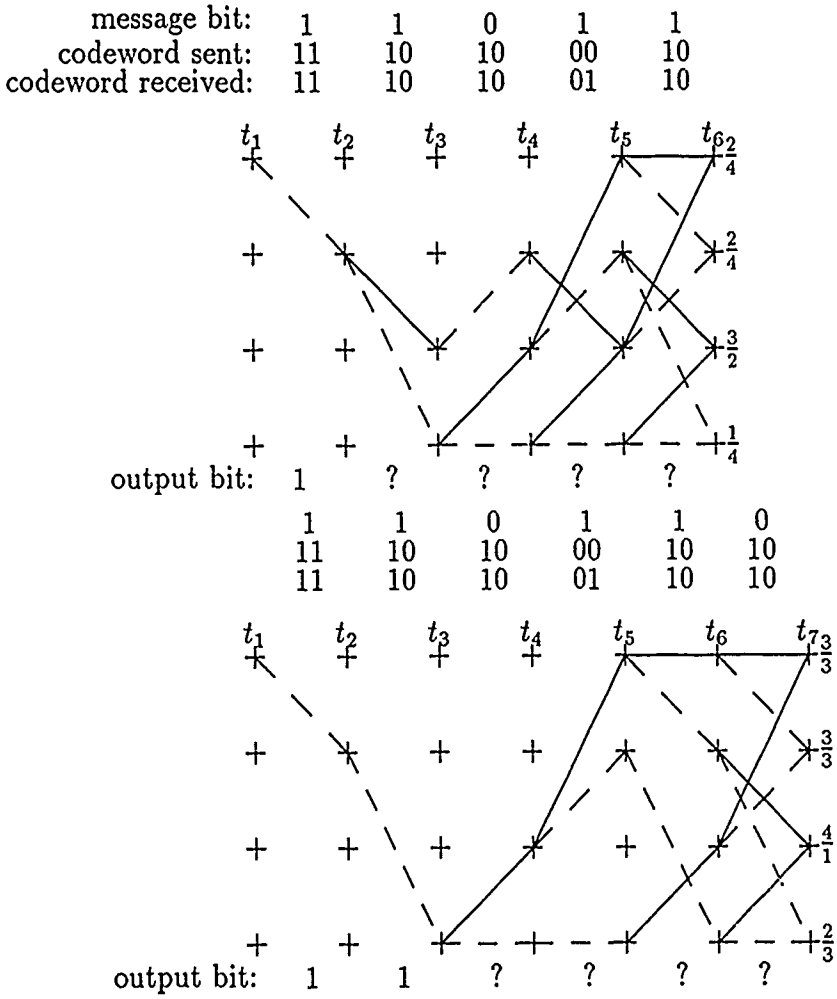


Figure 3.8. Surviving Paths at t_6,t_7

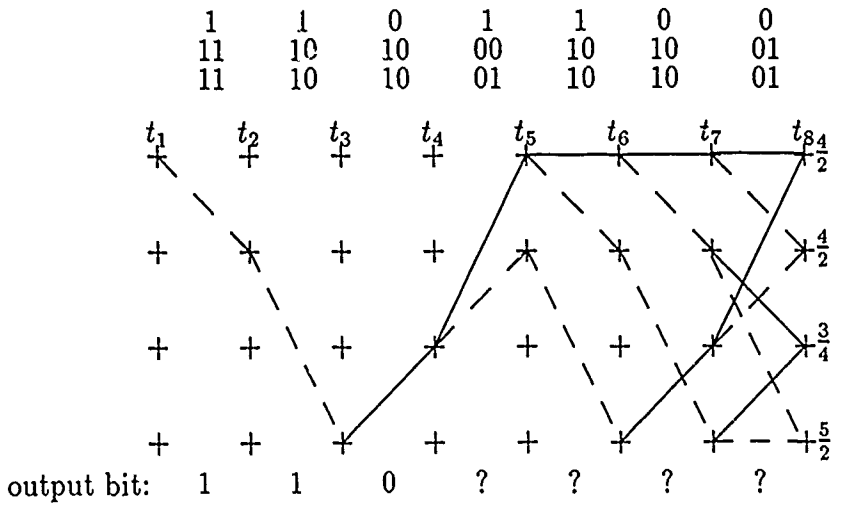


Figure 3.9. Surviving Paths at t_8

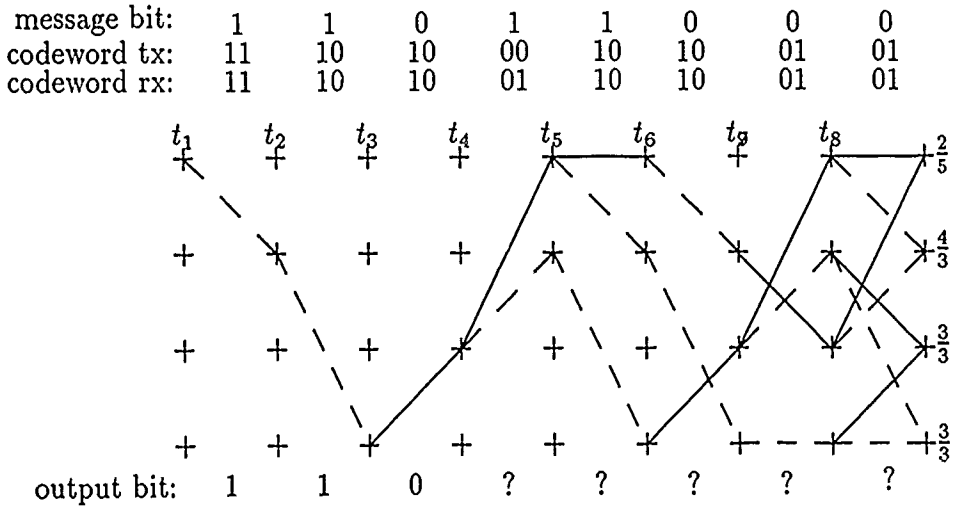


Figure 3.10. Surviving Paths at t_9

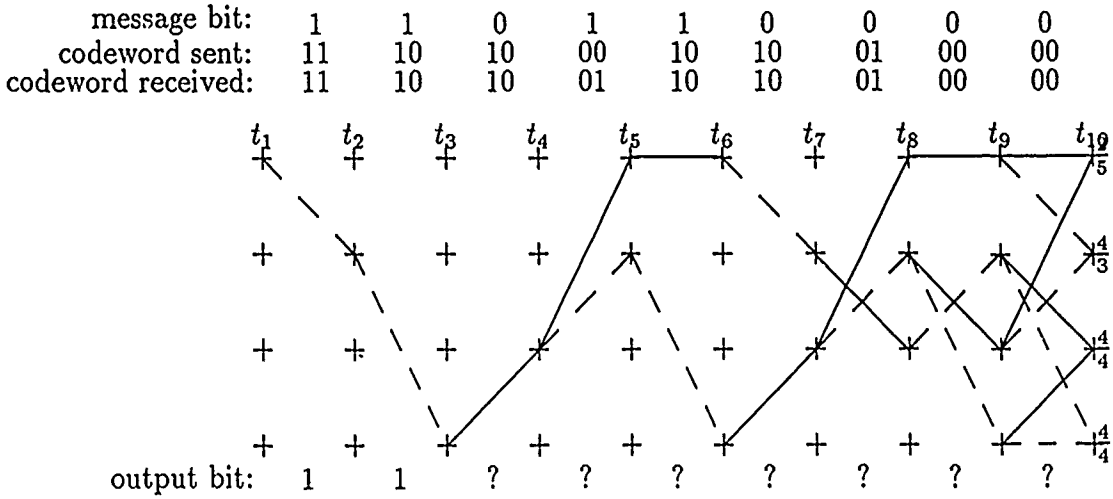


Figure 3.11. Surviving Paths at t_{10}

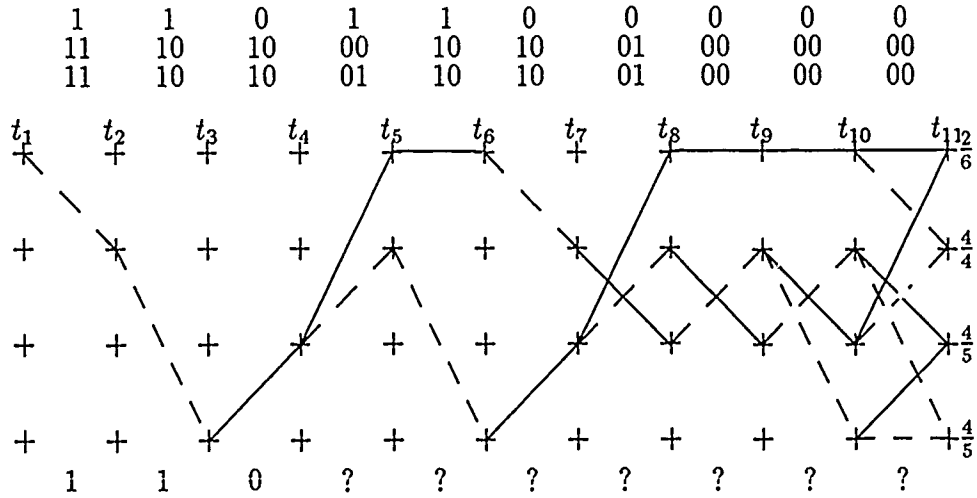


Figure 3.12. Surviving Path at time t_{11}

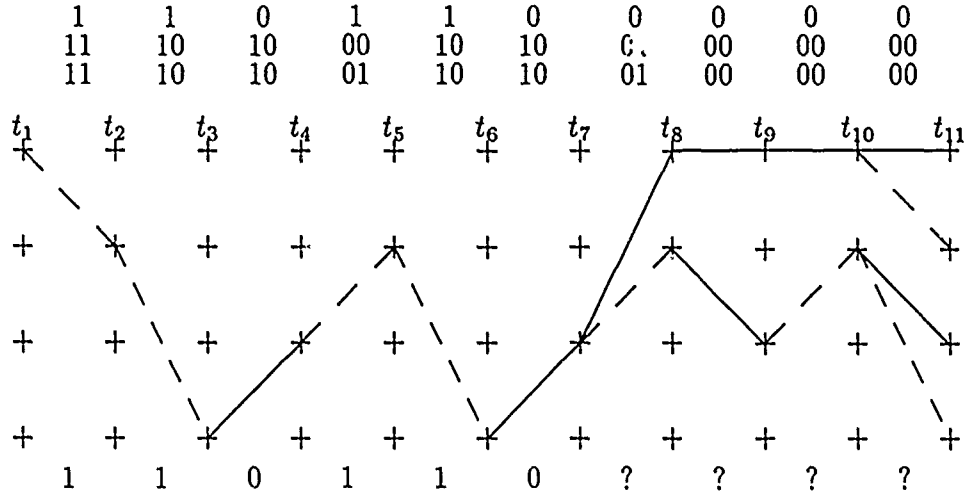


Figure 3.13. Surviving Path: Message Decoded

3.4 Error Performance

As mentioned in Section IV, the atmospheric channel is not memoryless. In fact, "the atmospheric optical channel has, however, a very long memory and a fade can cause an abnormally large number of erors that span thousands of consecutive received channel bits (12:993)." Interleaving is a common technique for overcoming the bursty errors that are caused by channel memory. Interleaving is a technique where the data bits are not transmitted successively. The first bit of a byte is sent, followed by the first bit of the second byte, until the first bit of the n^{th} byte is sent. N is defined as the interleaving depth. After the first bit of n bytes is sent, the second bit is sent, until all bits are sent. This is illustrated in Figure 3.14. After interleaving to a sufficient depth, a burst of errors will no longer effect more than one bit during the coherence time of the channel, thus making the channel appear memoryless. Unfortunately, as Davidson points out:

> Since the duration of the fades is random, no single maximum interleaving depth can be used to render the channel completely memoryless. Furthermore, at high source data rates, interleaving depths that correspond to time separations of 1 ms between successive bits of a codeword

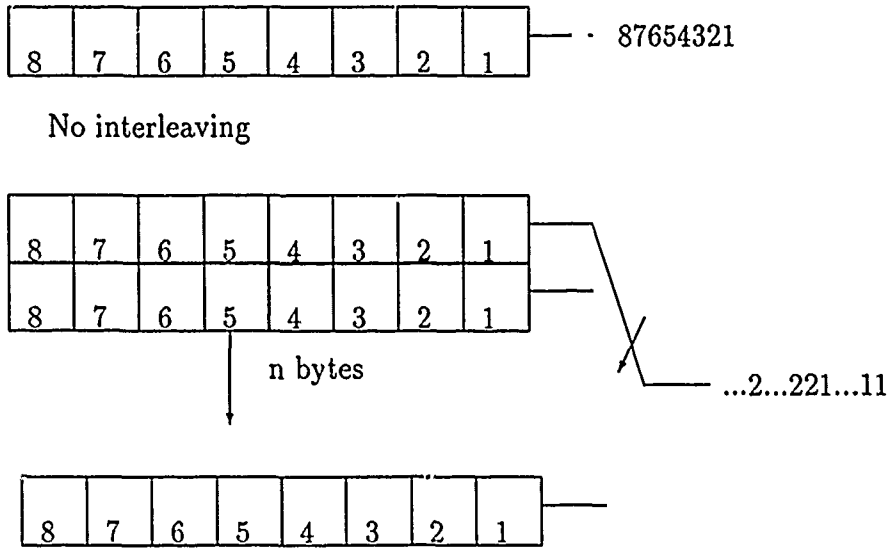


Figure 3.14. Interleaving a Data Stream

require the encoder and the decoder to contain very large amounts of memory (12:993).

Obviously the tradeoff is interleaving depth for data rate. The specific interleaving depth and data rate assumed in this thesis are discussed in Chapter IV.

As with any digital communications link, when a convolutional coding scheme is used, the probability of bit error (P_b) needs to be specified. P_b is a function of the code used, the decoding method, and the probability of error in the uncoded channel. The tool used to account for all these factors is the transfer, or generating function for the convolutional code, T(D,N). T(D,N) is obtained from a modified version of the state diagram given in Figure 3.5. This modified state diagram is shown in Figure 3.15. The *a* or 00 state, can be considered a beginning state and an ending state, is split up into two states, *a* and *e*, to explicitly show this. The state transition paths are labeled with the output codewords and a function of D, where the power of D is the Hamming distance between that codeword and the all zero codeword. According to Sklar,

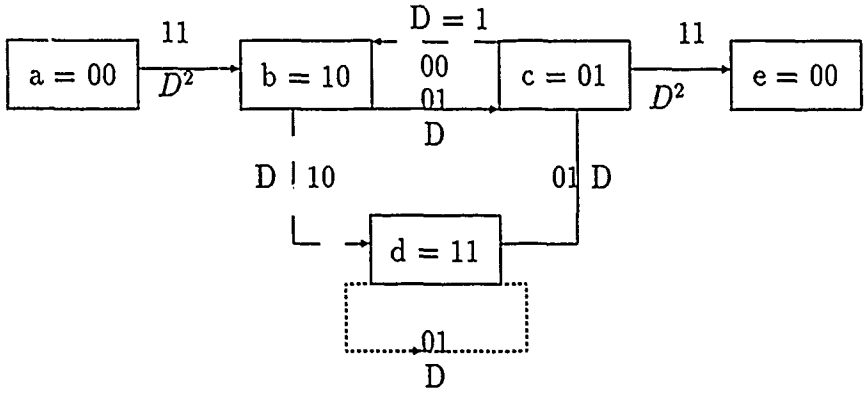


Figure 3.15. Modified State Diagram

> Because a convolutional code is a group or linear code (28), there is no loss in generality in simply finding the minimum distance between each of the codeword sequences and the all-zeros sequence (40:338).

N is a placeholder that keeps track of the number of transitions that were caused by an input bit of 1. The power of N is increased by one each time a state transition is caused by a 1. The state equations for this modified state diagram can now be written as:

$$\begin{aligned} X_b &= ND^2X_a + NX_c \\ X_c &= DX_b + DX_d \\ X_d &= NDX_b + NDX_d \\ X_e &= D^2X_c \end{aligned} \tag{3.1}$$

These state equations can be solved fairly easily in this case since the number of state equations is given by 2^{K-1}, which is 4 in this case where $K = 3$. If $K \geq 4$ it is more effective to use matrix methods to solve the state equations. Two approaches have been taken, differing only in terminology, not in logic. Sklar chose to leave X_a in the equations and solve for X_e/X_a as the transfer function (40:340). Clark and Cain treated the input as a unit impulse, setting $X_a = 1$ and then solved for X_e.

Using Clark and Cain's approach, the equations can be written in matrix form as:

$$\boldsymbol{X} = \boldsymbol{A}\boldsymbol{X} + \boldsymbol{X}_o \tag{3.2}$$

where $\boldsymbol{X}$ and $\boldsymbol{X}_o$ are $2^{K-1} \times 1$ vectors and $\boldsymbol{A}$ is a $2^{K-1} \times 2^{K-1}$ matrix. For the case of the $K = 3$ encoder in Figure 3.2:

$$\boldsymbol{A} = \begin{bmatrix} 0 & N & 0 & 0 \\ D & 0 & D & 0 \\ ND & 0 & ND & 0 \\ 0 & D^2 & 0 & 0 \end{bmatrix}$$

and

$$\boldsymbol{X}_o = \begin{bmatrix} ND^2 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Solving for $\boldsymbol{X}$, $\boldsymbol{X} = (\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{X}_o$. The transfer function T(D,N) is then given by $\boldsymbol{X}_4$ in this case, or by $\boldsymbol{X}_{2^{K-1}}$ in general.

Viterbi has shown that P_b can be bounded using the transfer function as shown in equation 3.3 (49:763).

$$P_b \leq \left. \frac{dT(D,N)}{dN} \right|_{N=1, D=D_0} \tag{3.3}$$

where D_0 is a function of the uncoded channel error probability. It would be nice to be able to determine the statistics of P_b exactly, unfortunately this can not be done since the statistics of the path metrics can not be evaluated in closed form (12:996). The following chapter will show how these equations for the bounds can be used to calculate P_b curves for the different coding schemes.

IV. Methodology

Davidson showed that for a medium level of turbulence, 5 msec of interleaving delay made the channel appear sufficiently memoryless. In this case, the soft-decision encoder's measured bit errors closely agree with the theoretical curves for a memoryless channel (12). Assuming a data rate of 10 kbps, the required interleaving depth would be 50, whereas if a slower rate, like 2 kbps, could be tolerated, then the required interleaving depth could be reduced to 10. These interleaving depths can be compared to those that would be required for perfect interleaving. If the time scale of the fading is assumed to be 1 second, which is typical for the atmosphere, then at 10kbps the interleaving depth would have to be 1000 and at 2 kbps it would have to be 200. These interleaving depths are impractical due to speed and memory constraints, making Davidson's results extremely valuable for being able to assume near memoryless conditions at substantially reduced interleaving depths.

Assuming sufficient interleaving, the generating function for the encoder can now be used with the appropriate D_o for predicting the bit error performance. Using the Poisson statistics for the photon arrival rates, Davidson showed that for a soft-decision, binary pulse-position modulation (BPPM) channel, D_o is computed to be

$$D_o(\chi) = e^{-(\sqrt{\lambda_s e^{2\chi}+\lambda_0}-\sqrt{\lambda_0})^2 T_0/2} \tag{4.1}$$

where λ_s and λ_0 are the average rates of arrival of the signal and background photons, respectively, T_0 is simply the bit duration time, and χ is the log-amplitude of the signal. In order to compare the plots generated in this thesis to Davidson's plots, this equation needs to be written in terms of $\bar{N}_s$ and $\bar{N}_0$ rather than λ_s, λ_0 and T_0. Bringing $T_0/2$ inside the radical, equation 4.1 can be rewritten as equation 4.2.

$$D_o(\chi) = e^{-\left(\sqrt{\frac{N_s e^{2\chi}+N_0}{2}}-\sqrt{\frac{N_0}{2}}\right)^2} \tag{4.2}$$

As discussed in Chapter II, scintillation effects cause χ to be normally distributed. To account for the scintillation effects, $Z(\chi)$ must be integrated over this normal density to obtain the final expression for D given in equation 4.3.

$$D_o = \int_{-\infty}^{\infty} \frac{e^{-\left(\sqrt{\frac{N_s e^{2\chi}+N_0}{2}}-\sqrt{\frac{N_0}{2}}\right)^2}}{\sqrt{2\pi\sigma_\chi^2}} e^{-\frac{\chi^2}{2\sigma_\chi^2}} d\chi \tag{4.3}$$

The integration in equation 4.3 was first done numerically using MathStation (32). Unfortunately, the resulting P_b's were three to four orders of magnitude higher than those reported by Davidson. Since χ is normally distributed, the first check was to ensure that the area above and below the numerical limits of integration were in fact negligible. As the range of integration was increased, the value of D_o decreased, verifying that the integration was being done incorrectly. The version of MathStation being used was determined to be designed for an older version of the SUN operating system. This seemed to a likely explanation for the incorrect integration. However, a new version of MathStation was received which was supposed to be compatible with present version of the operating system. Unfortunately, this new version gave the same results as the old version. The final option, which fortunately was successful, was to use the same twenty point gauss-hermite integration that Davidson used. A C program was written to perform the summation. It is included in Appendix F. This program was checked on a zero mean, unit variance, normal distribution. The result was within .001% of the correct result of one. After multiplying the normal distribution by $D_o(\chi)$ as given by equation 4.2 and integrating, a arithmetic underflow was encountered. Since the problem seemed to be ocurring in the exponential, equation 4.3 was rewritten using equation 4.1 with $T_o = 1$, instead of equation 4.2. The probability of bit error curves are then obtained by substituting equation 4.3 into the generating function and plotting over a range of values for N_s and specific values of N_0 and σ_χ.The resulting P_b's were within one order of magnitude of Davidson's results.

Chapter III showed the derivation of the state equations for the case of $K = 3$. Writing out a state diagram for cases where $K \geq 4$ gets extremely cumbersome. The method used in this thesis was to write out all possible states twice and then put either a 1 or a 0 in front, representing an input bit of 1 or 0. The output codeword was then determined by the taps and the contents of the register. The method is illustrated in equation 4.4.

$$\begin{array}{llcll} 000 & X_a & \Rightarrow & X_a & 00 \\ 100 & X_e & \rightarrow & X_c & 11 \\ 001 & X_b & \Rightarrow & X_e & 11 \\ 101 & X_b & \rightarrow & X_c & 00 \\ 010 & X_c & \Rightarrow & X_b & 01 \\ 110 & X_c & \rightarrow & X_d & 10 \\ 011 & X_d & \Rightarrow & X_b & 01 \\ 111 & X_d & \rightarrow & X_d & 10 \end{array} \tag{4.4}$$

The double arrow represents a transition caused by a zero. Using these equations, the state equations can be written as:

$$\begin{aligned} X_b &= DX_c + DX_d \\ X_c &= ND^2X_a + NX_b \\ X_d &= NDX_c + NDX_d \\ X_e &= D^2X_b \end{aligned} \tag{4.5}$$

These state equations are the same as those given in Chapter III except for the b and c rows being swapped. This is simply a result of the b and c states being swapped from 10 to 01 and vice versa.

The codes examined in this thesis are derivatives of a code developed by Massey and Costello. They developed an algorithm to determine generating polynomials

that would produce non-catastrophic, nonsystematic, optimal codes. One of the codes they discovered has a constraint length of 48 and is represented octally by 533,533,676,737,355,3 and 733,533,676,737,355,3. Fortunately, they also showed that shorter codes with these same desirable characteristics could be obtained by simply dropping elements off of the right end until the desired constraint length is reached (30). Interestingly, and conveniently, the K = 3 code obtained in this manner is the same as the example mentioned in Chapter III. This provided an excellent means for verifying the results obtained, as they could be checked against the work done by Davidson. The method could then be extended to the longer constraint codes with reasonable confidence as to the accuracy of the results. The extension is straightforward conceptually, however, the number of calculations required for an exact solution becomes immense. In order to obtain an exact solution, the matrix (I-A) must be inverted symbolically in terms of D and N. This was done on a Sun 3 workstation using Mathematica (31), a software package that can handle symbolic mathematics. For the $K \leq 5$ case, the inversion was done with no appreciable time delay. Unfortunately, for $K = 6$, after a week, the computer had not finished the inversion. Obviously, a numerical, approximate solution was needed. Viterbi states in his article on convolutional codes that

> ... a generating function calculation is basically just a matrix inversion, which can be performed numerically for a given value of D. The derivative at N = 1 can be upper bounded by evaluating the first difference
>
> $$[T(D,1+\epsilon) - T(D,1)]/\epsilon \qquad (4.6)$$
>
> for small ϵ (49:761)

The P_b curve in Figure 4.1 was produced using $P_b \leq \frac{dT(D,N)}{dN}\Big|_{N=1,D=D_o}$ for $K = 3$, where, $\frac{dT(D,N)}{dN}\Big|_{N=1,D=D_o} = D_o^5/(1-2D_o)^2$. Figure 4.2 shows the curve produced by the the numerical method with $\epsilon = .01$. These two curves are overlaid in Figure 4.3. The actual data that was used to generate the plots is shown in Ta-

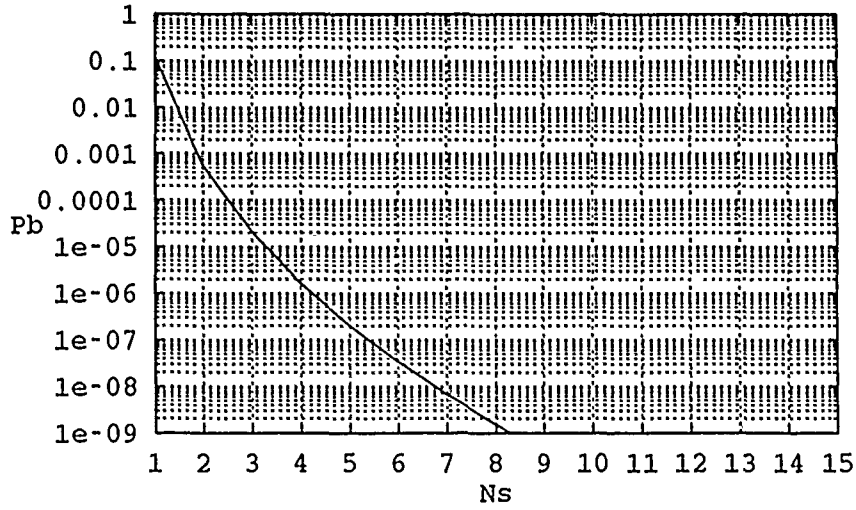


Figure 4.1. P_b Curves from Exact T(D,N)

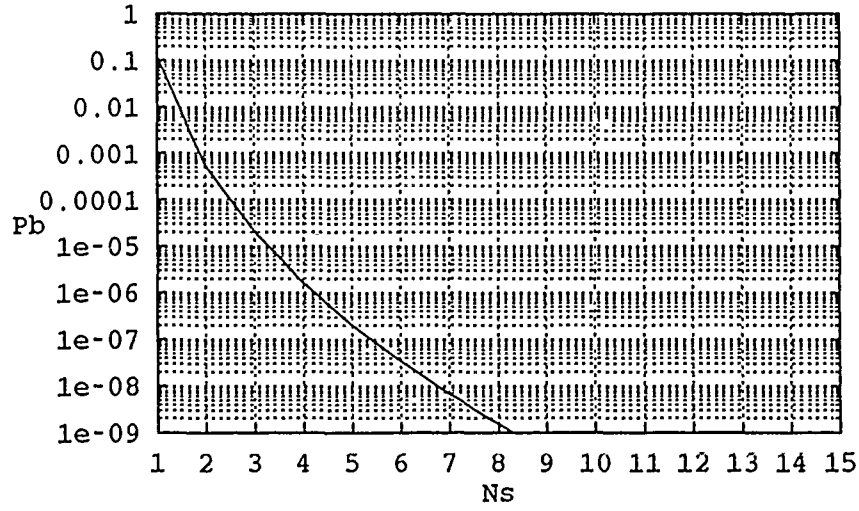


Figure 4.2. P_b Curves from Approximate Method

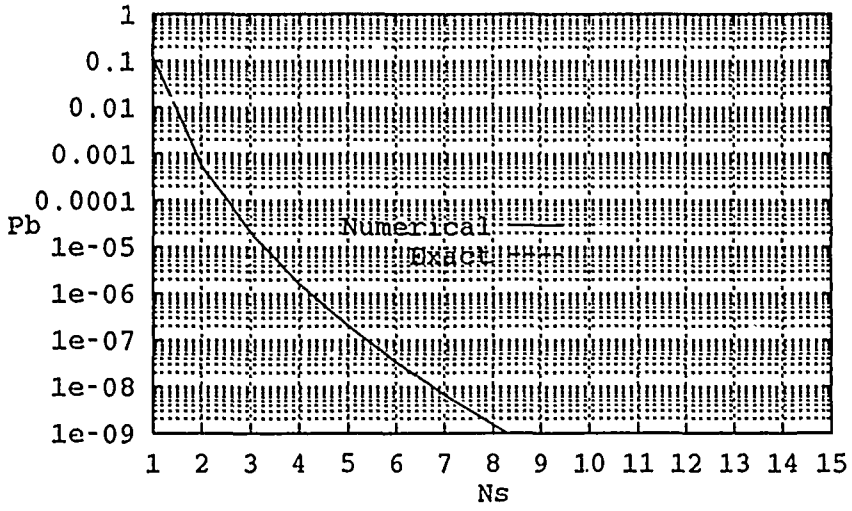


Figure 4.3. Overlaid Curves from both Methods

Table 4.1. Comparison of Exact and Numerical Methods for Finding P_b

N_s	$P_b \leq \frac{dT(D,N)}{dN}\Big\|_{N=1,D=D_o}$	$P_b \leq \frac{T(D,1+\epsilon)-T(D,1)}{\epsilon}$	% Error
1	$1.219574370237228\ 10^{-1}$	$1.257684126095466\ 10^{-1}$	3.10
2	$5.408651034564673\ 10^{-4}$	$5.440526112330407\ 10^{-4}$	0.58
3	$2.068339667884289\ 10^{-5}$	$2.073866818209250\ 10^{-5}$	0.26
4	$1.625352476271095\ 10^{-6}$	$1.627815938605986\ 10^{-6}$	0.15
5	$2.010669397505719\ 10^{-7}$	$2.012617133662639\ 10^{-7}$	0.09
6	$3.339719190100789\ 10^{-8}$	$3.341940128714635\ 10^{-8}$	0.06
7	$6.676329368437486\ 10^{-9}$	$6.679512188932890\ 10^{-9}$	0.05
8	$1.507209044081476\ 10^{-9}$	$1.507738692119392\ 10^{-9}$	0.04
9	$3.725211292170222\ 10^{-10}$	$3.726196035074627\ 10^{-10}$	0.03
10	$9.956548639013677\ 10^{-11}$	$9.958562668240980\ 10^{-11}$	0.02
11	$2.867396086188684\ 10^{-11}$	$2.867847057830152\ 10^{-11}$	0.02
12	$8.891537688029467\ 10^{-12}$	$8.892641949738258\ 10^{-12}$	0.01
13	$2.965991554500035\ 10^{-12}$	$2.966286851056265\ 10^{-12}$	0.01
14	$1.061641638028914\ 10^{-12}$	$1.061727606677203\ 10^{-12}$	0.01
15	$1.648835809525375\ 10^{-13}$	$1.648927667019529\ 10^{-13}$	0.01

ble 4.1. The percent error using the numerical approximation, was calculated as $(\frac{exact-numerical}{exact})100\%$. It can be seen in the plots and from the data that the numerical approximation is a good one for $N_s \geq 4$. For $N_s \leq 4$ neither method provides realistic results as P_b's greater than one and even less than zero were obtained. Davidson's plots show the curves asymptotically approaching 1 for $N_s \approx 2$. He makes no mention of the fact that the numbers do not make practical sense for this region of low SNR.

This method can now be used to calculate the P_b curves for all constraint lengths ≤ 7. The D parameter from equation 4.3 was calculated for N_s ranging from 1 to 15. The D parameter was then used in a MATRIXx (33) program that evaluated equation 4.6. The MATRIXx command files that were written are shown in Appendix A. For $K = 8$, the huge number of calculations and resulting roundoff errors made the matrix appear singular within MATRIXx's precision. However, based on the results obtained from $K = 6$ and $K = 7$, as shown in Figures 5.1 through 5.9, the curves start getting closer and closer with increasing K. Since 7 is a common constraint length, it seemed like a good point to stop based on diminishing returns.

V. Results

The results of Chapter IV's calculations are presented here. Tables of the raw data are given in the appendices, while the P_b curves, are presented in this chapter.

The MATRIXx command files from Appendix A were used to analyze codes with constraint lengths from 3 to 7. The D parameter that was used as an input was calculated for no turbulence, ($\sigma^2_{ln\lambda} = .01$), low turbulence, ($\sigma^2_{ln\lambda} = .2$) and for medium turbulence, ($\sigma^2_{ln\lambda} = .6$), as well as for three different background light levels, $N_o = .1$, 1, and 10 photons/bit. $\sigma^2_{ln\lambda}$ is a parameter Davidson uses to describe the strength of the turbulence. He defines $\lambda(t)$ as the received signal, by:

$$\lambda_s(t)e^{2\chi(t)} \tag{5.1}$$

He defines λ_s as the "information bearing intensity modulation impressed on the laser beam at the transmitter (12:994)." $\sigma^2_{ln\lambda}$ is then defined as:

$$\sigma^2_{ln\lambda} = E\left\{(ln\lambda(t))^2\right\} - E^2\left\{ln\lambda(t)\right\} \tag{5.2}$$

Using equation 5.1 and assuming that $\lambda_s(t)$ is deterministic, equation 5.2 can be rewritten as:

$$\begin{aligned} \sigma^2_{ln\lambda} &= Var\left\{ln\lambda(t)\right\} \\ &= Var\left\{ln\lambda_s(t) + 2\chi(t)\right\} \\ &= Var\left\{2\chi(t)\right\} \\ &= 4\sigma^2_\chi \end{aligned} \tag{5.3}$$

Tables of the D parameter are given in Appendix C. Tables of the resulting P_b data are given in Appendix D. Although the calculations were made in each case for N_s ranging from 1 to 15, in each case there was a certain $N_{s_{min}}$ below which the signal to

noise ratio was too low for this method to work. Below $N_{s_{min}}$, P_b could go negative or even above 1. Data points for $N_s \leq N_{s_{min}}$ were not plotted. They were retained in the tables of data found in Appendix D. $N_{s_{min}}$ was determined for each case as the point where the curve broke continuity with decreasing N_s.

When the results of these calculations are compared to Davidson's results, there is a difference of $\approx$ one half of an order of magnitude. The cause for this difference is the evaluation of equation 4.3. The evaluation of equation 4.3 is very dependent on the method of integration, and the precision of the calculator. The goal of this thesis was to provide the communications engineer a means of determining the coding gain available from longer constraint length codes. The plots in this chapter provide that. The absolute value of P_b is not known anyway since these are only bounds on the error. The atmosphere is very difficult to model with any degree of reliability. The ability of these curves to accurately represent the correct value of P_b is dependent upon the level of turbulence, which is measured with σ_χ^2. σ_χ^2 is directly proportional to $C_{n'}^2$ which can vary 3 to 4 orders of magnitude.

Another factor that is somewhat hidden in this analysis, is the dependence of P_b on path length. This dependence is accounted for in σ_χ^2. Throughout this thesis a certain σ_χ^2 has been assumed. σ_χ^2 can be calculated from $C_{n'}^2$ (9:36) as:

$$\sigma_\chi^2 = .031 k^{7/6} L^{11/6} C_{n'}^2 \tag{5.4}$$

where k is still the wavenumber, which can be given as ω/c or $2\pi/\lambda$, and L is the pathlength. Unfortunately, equation 5.4's validity breaks down as the pathlength increases or $C_{n'}^2$ increases. Davidson alluded to this problem in his discussion of the data obtained for $\sigma_{ln\lambda}^2 \approx .6$ (12:999).

Even with these limitations a lot of information is shown in these plots. Figures 5.1 through 5.9 show that P_b can be varied over 5 orders of magnitude with the use of 5 constraint lengths and a fairly small range of N_s. For example, Figure 5.4

shows that for N_s ranging from 5 to 10, a dynamic range of only 3dB, P_B can be varied from $5 \cdot 10^{-4}$ with K = 3 at $N_s = 5$ to less than 10^{-9} with K = 7 and $N_s = 10$. Figures 5.3, 5.6 and 5.9 show that if the background light gets too high, turbulence does not matter anymore. Inordinately large values of N_s are required to get P_b low enough to be of any use. This is shown for constraint lengths of 3,5, and 7 in Figures 5.11 and 5.12 respectively.

Another important consideration is demonstrated in these plots. When selecting codes, the last shift register should be tapped. Otherwise, the encoder does not produce a maximal length code and the P_b is the same as that obtained from a code of the same length as the length from the first register in the code to the last register that has a tap. This is shown in Figures 5.1 through 5.9. The P_b curves for $K = 3$ is the same as $K = 4$ and the curves for $K = 6$ overlays $K = 7$. Seven is a common constraint length for a convolutional code, unfortunately for this particular code, there is no performance gain over a constraint length of six.

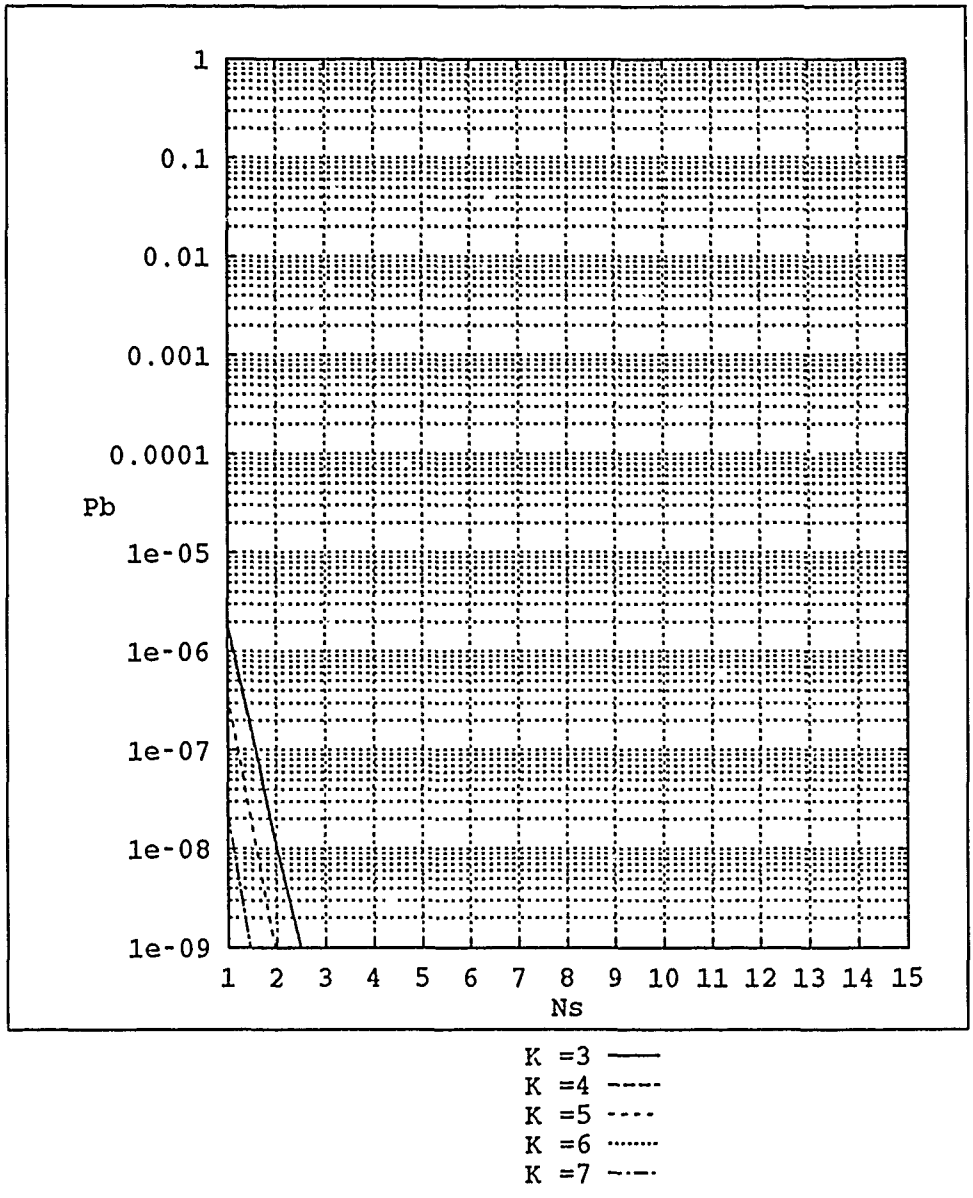


Figure 5.1. P_b for K = 3 thru 7 With No Turbulence, $N_o = .1$

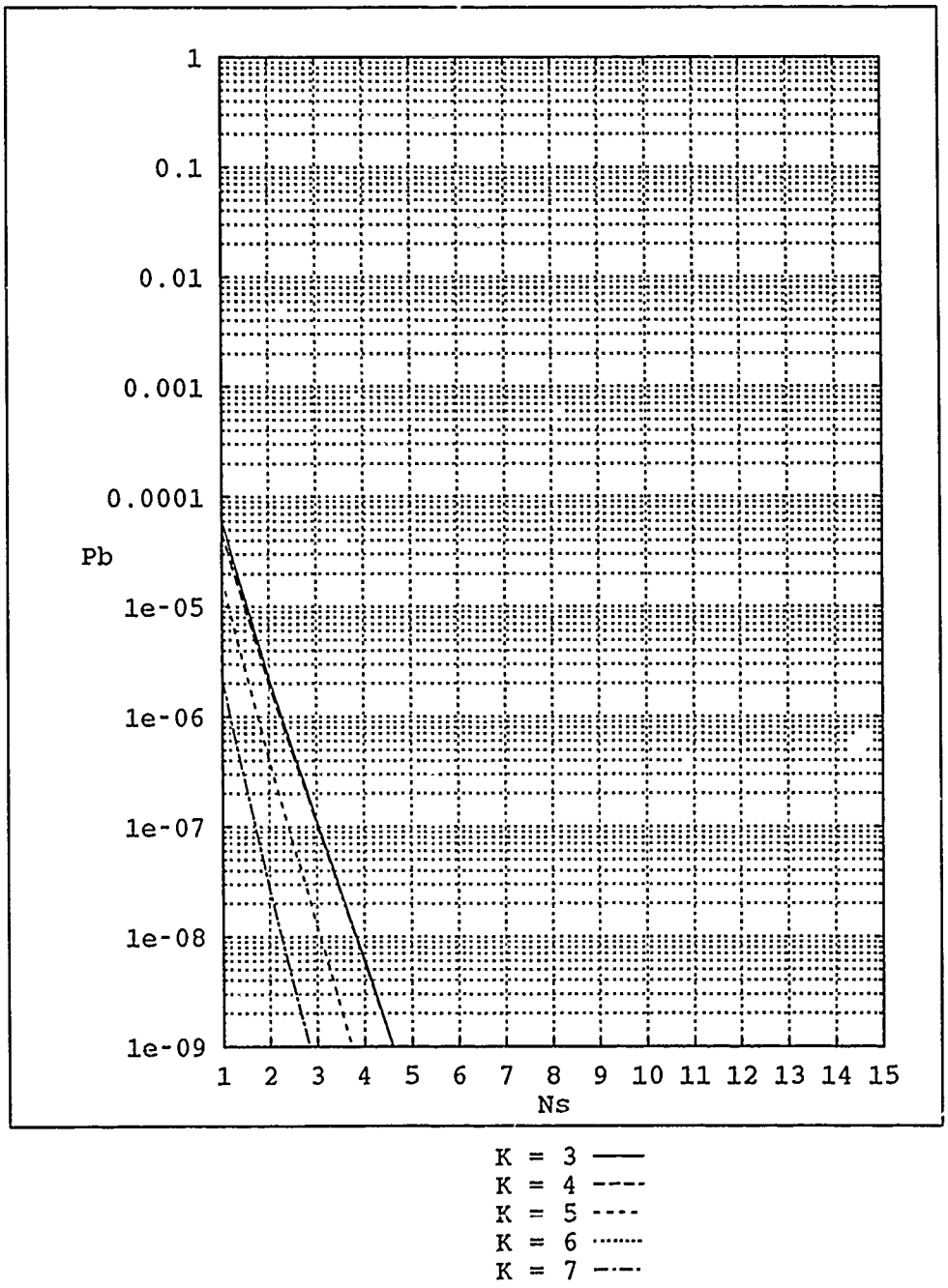


Figure 5.2. P_b for K = 3 thru 7 With No Turbulence, $N_o = 1$

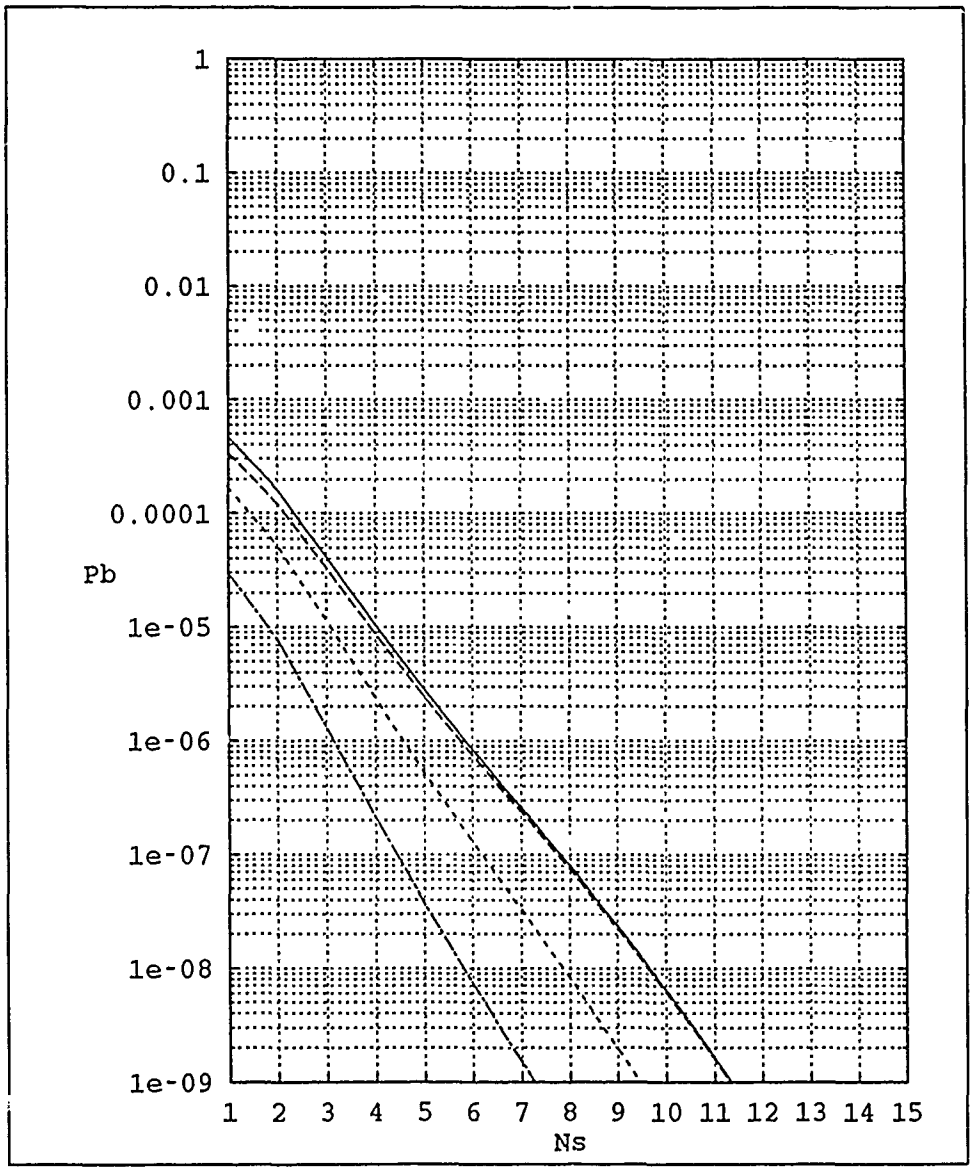


Figure 5.3. P_b for K = 3 thru 7 With No Turbulence, $N_o = 10$

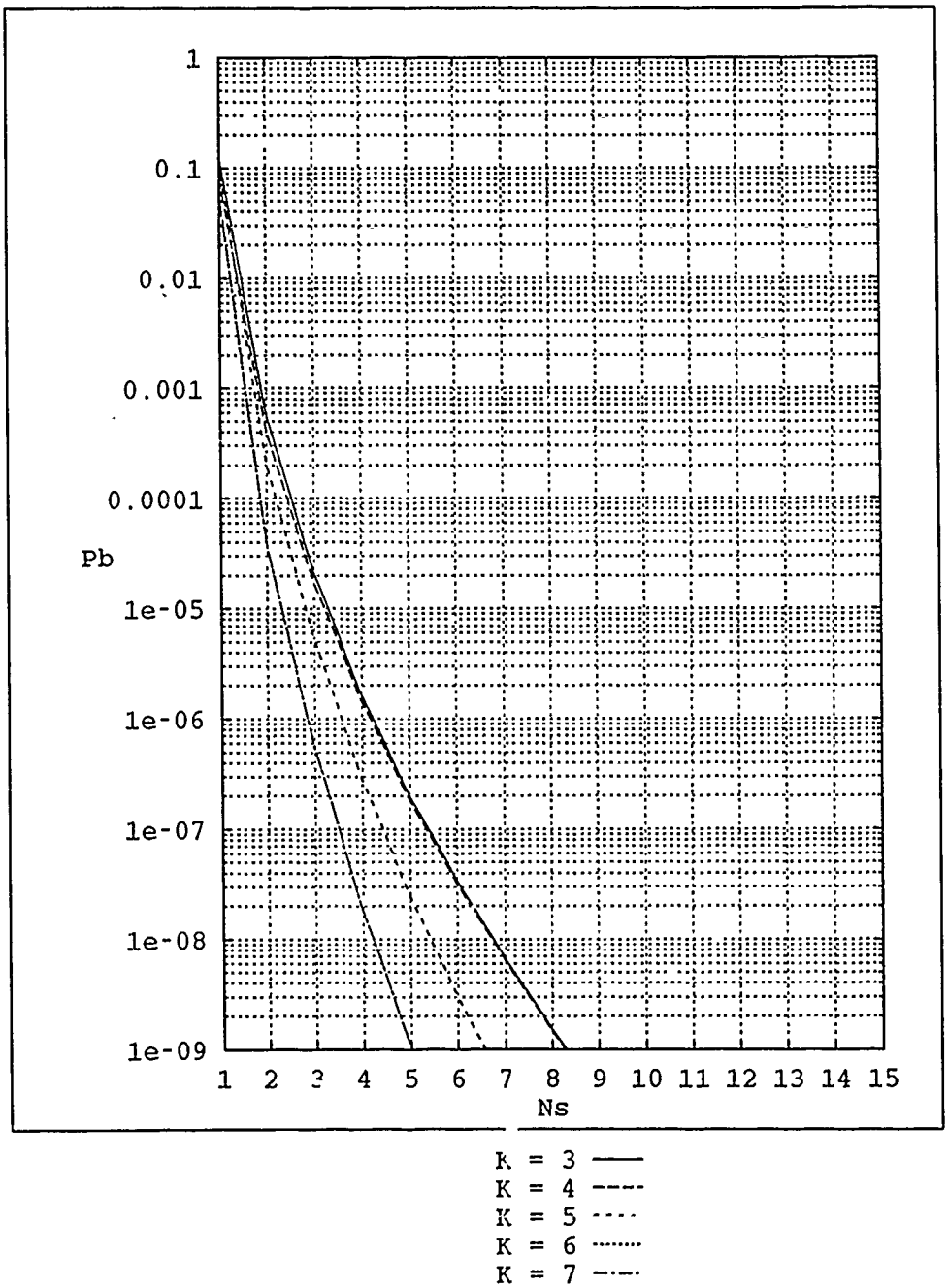


Figure 5.4. P_b for K = 3 thru 7 With Low Turbulence, $N_o = .1$

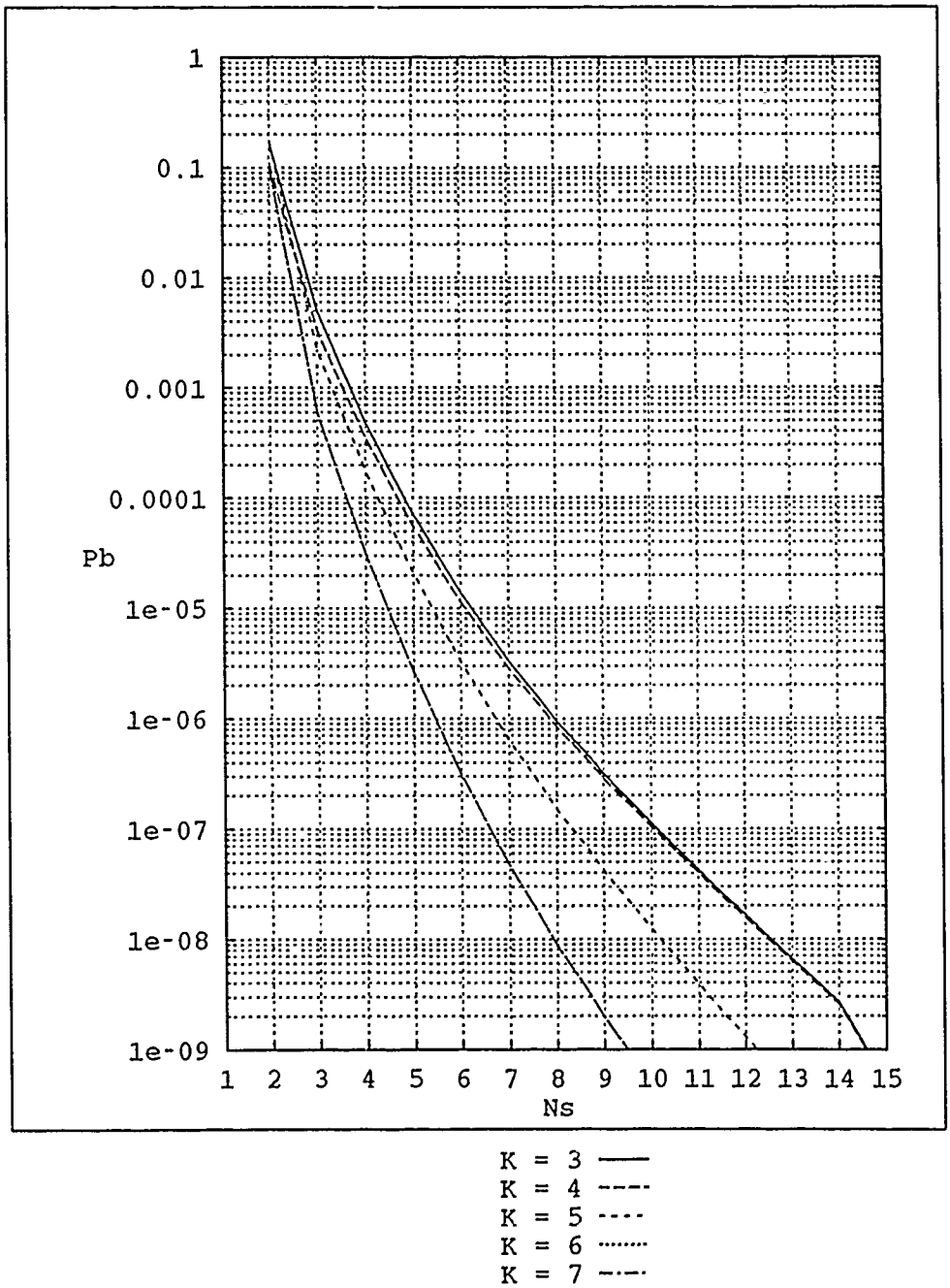


Figure 5.5. P_b for K = 3 thru 7 With Low Turbulence, $N_o = 1$

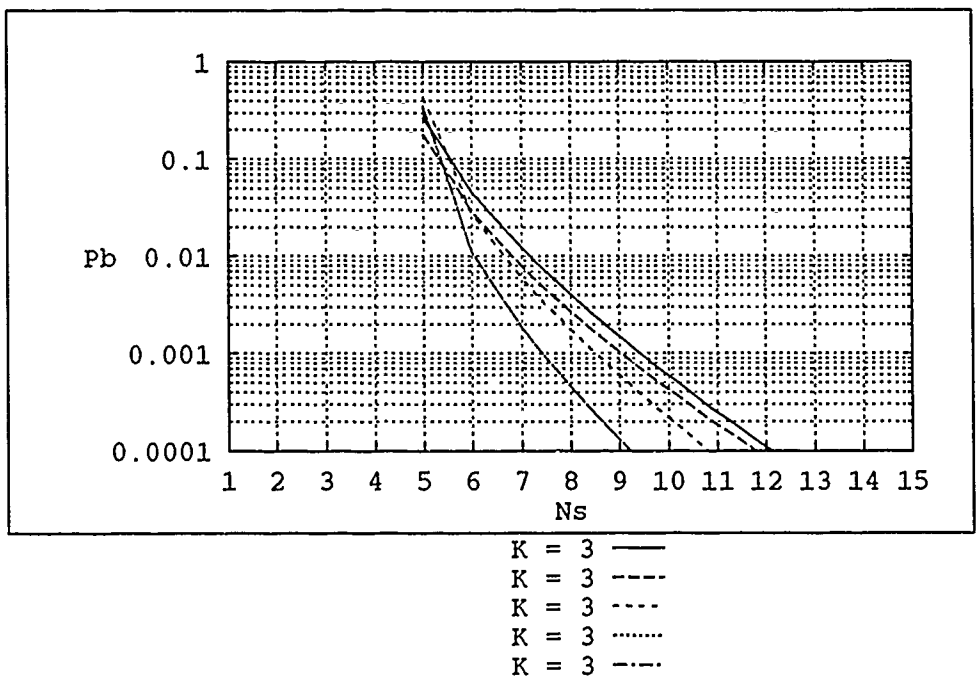


Figure 5.6. P_b for K = 3 thru 7 With Low Turbulence, $N_o = 10$

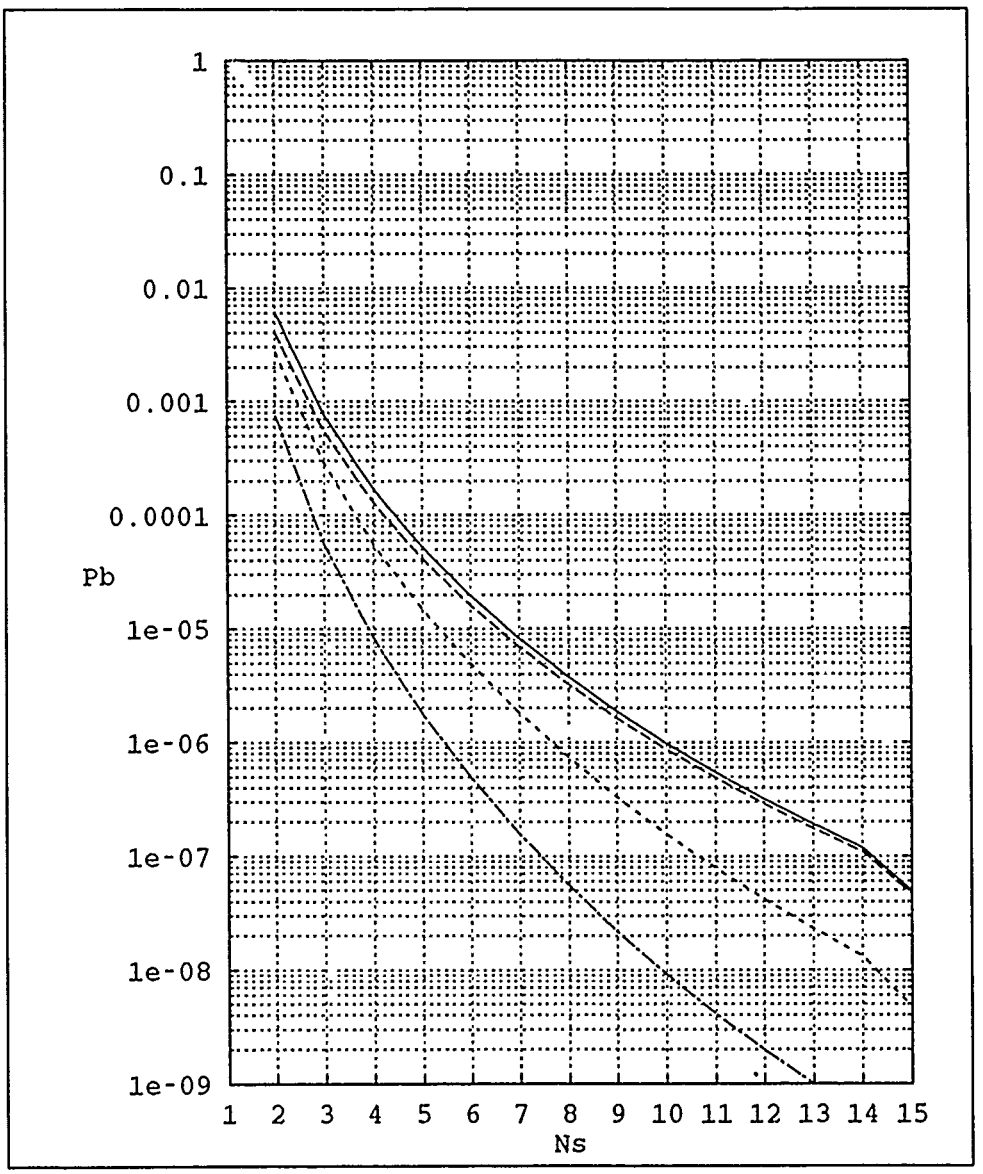


Figure 5.7. P_b for K = 3 thru 7 With Medium Turbulence, $N_o = .1$

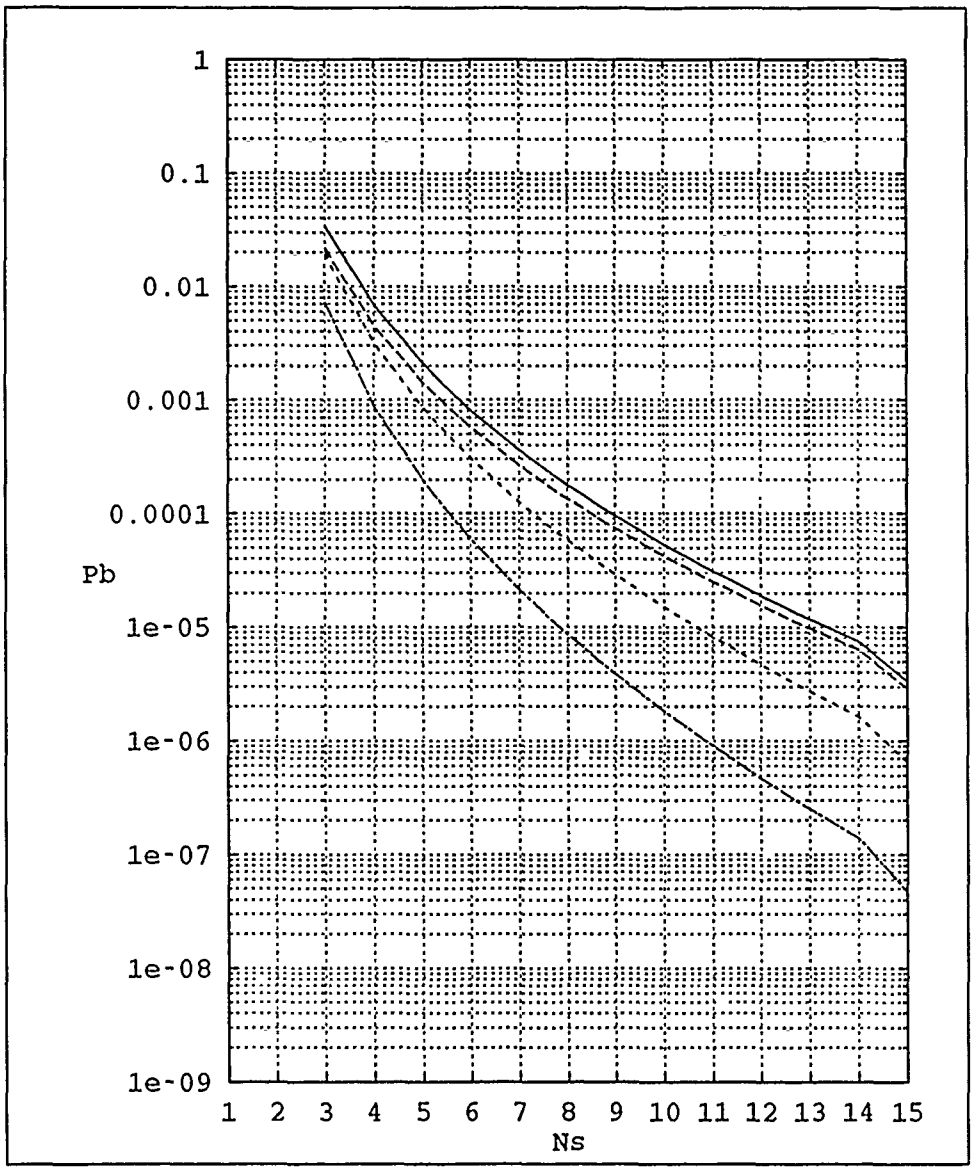


Figure 5.8. P_b for K = 3 thru 7 With Medium Turbulence, $N_o = 1$

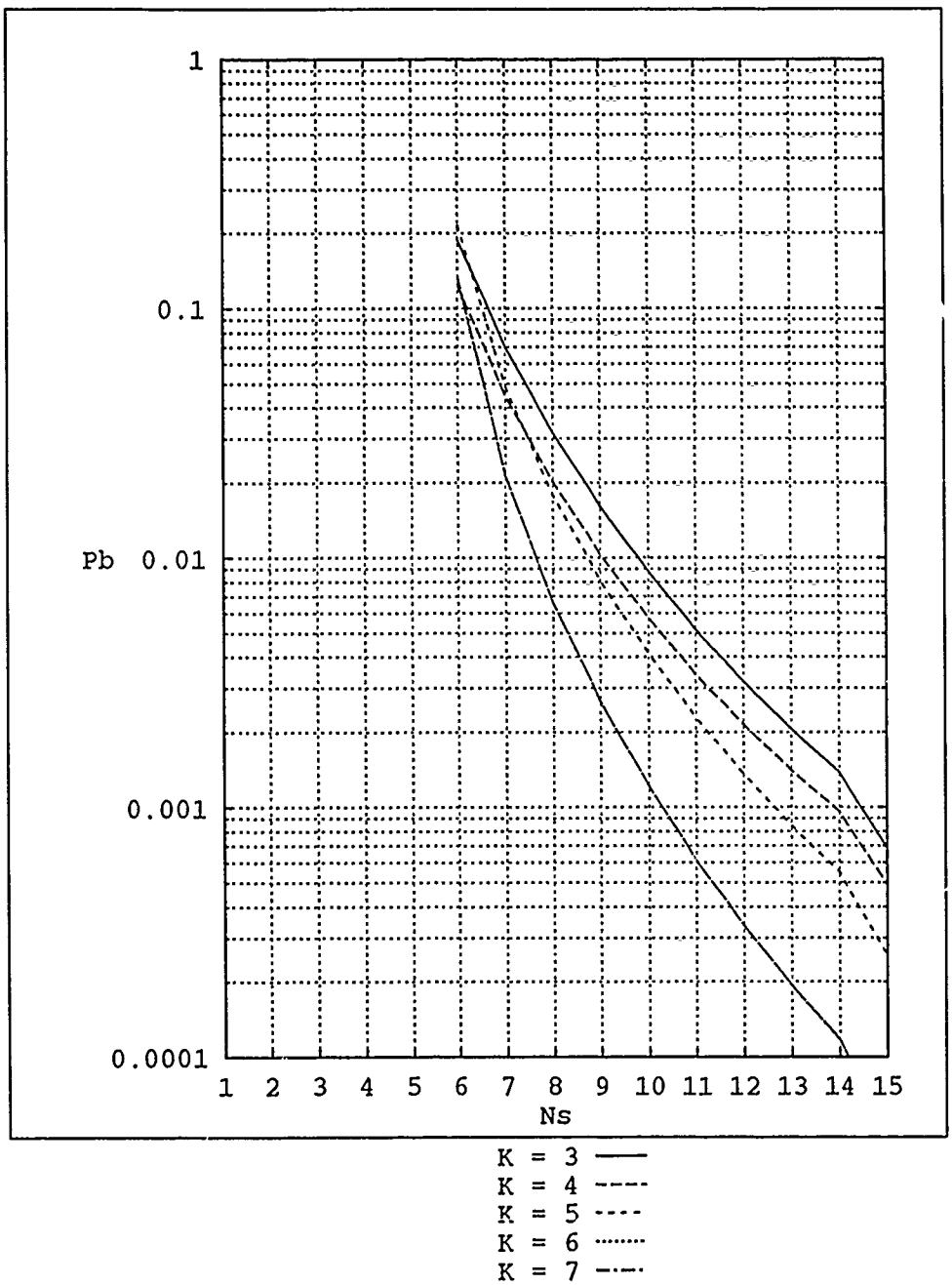


Figure 5.9. P_b for K = 3 thru 7 With Medium Turbulence, $N_o = 10$

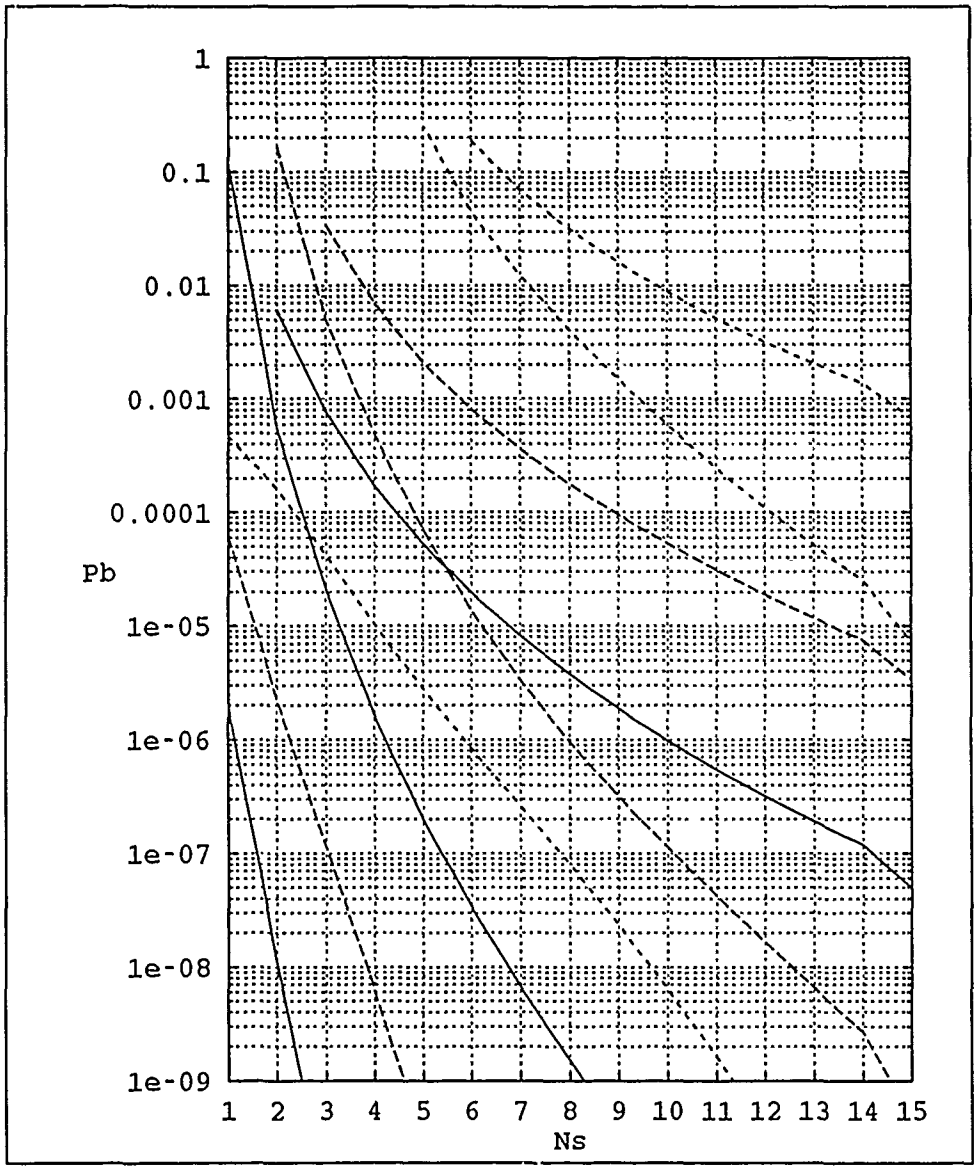


Figure 5.10. Comparing Turbulence and Background Effects for K = 3

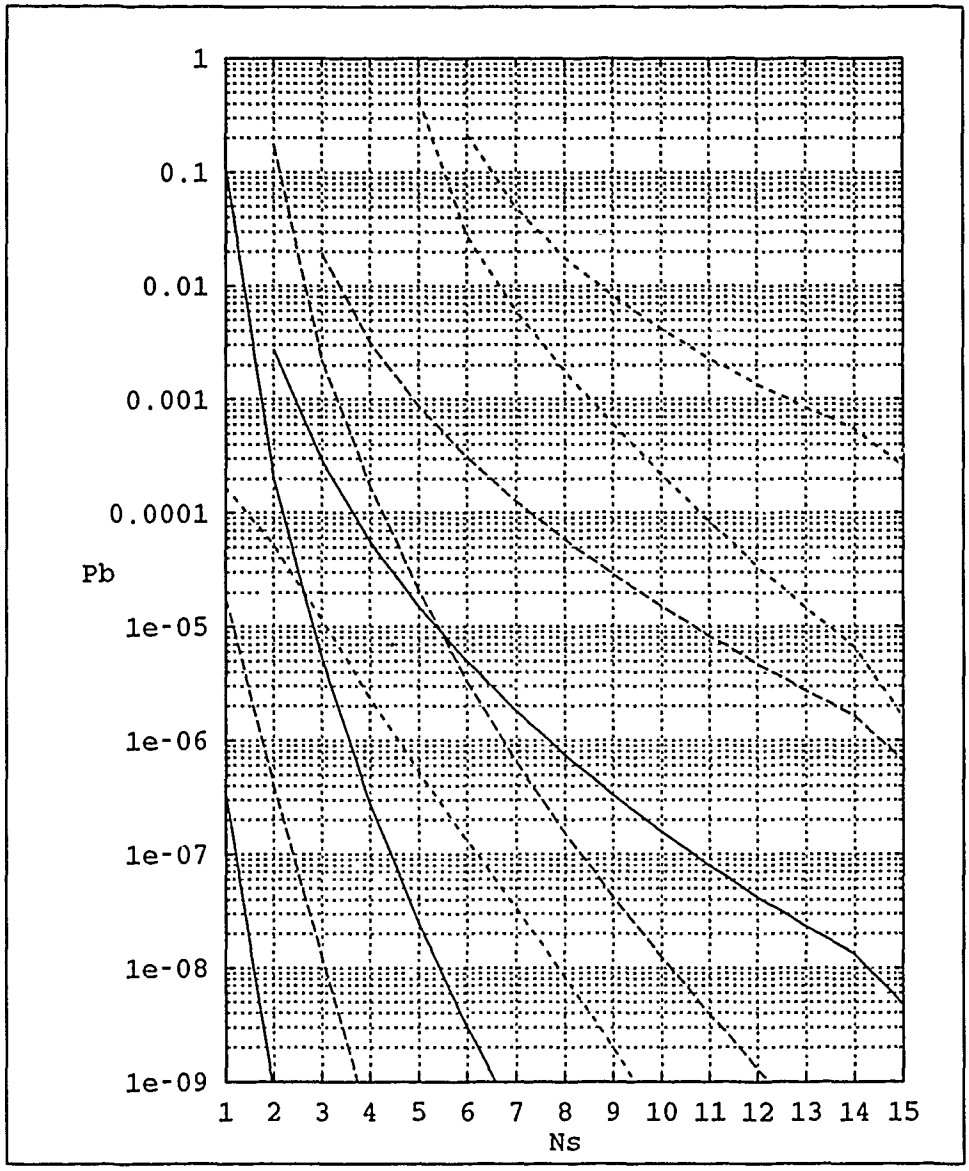


Figure 5.11. Comparing Turbulence and Background Effects for K = 5

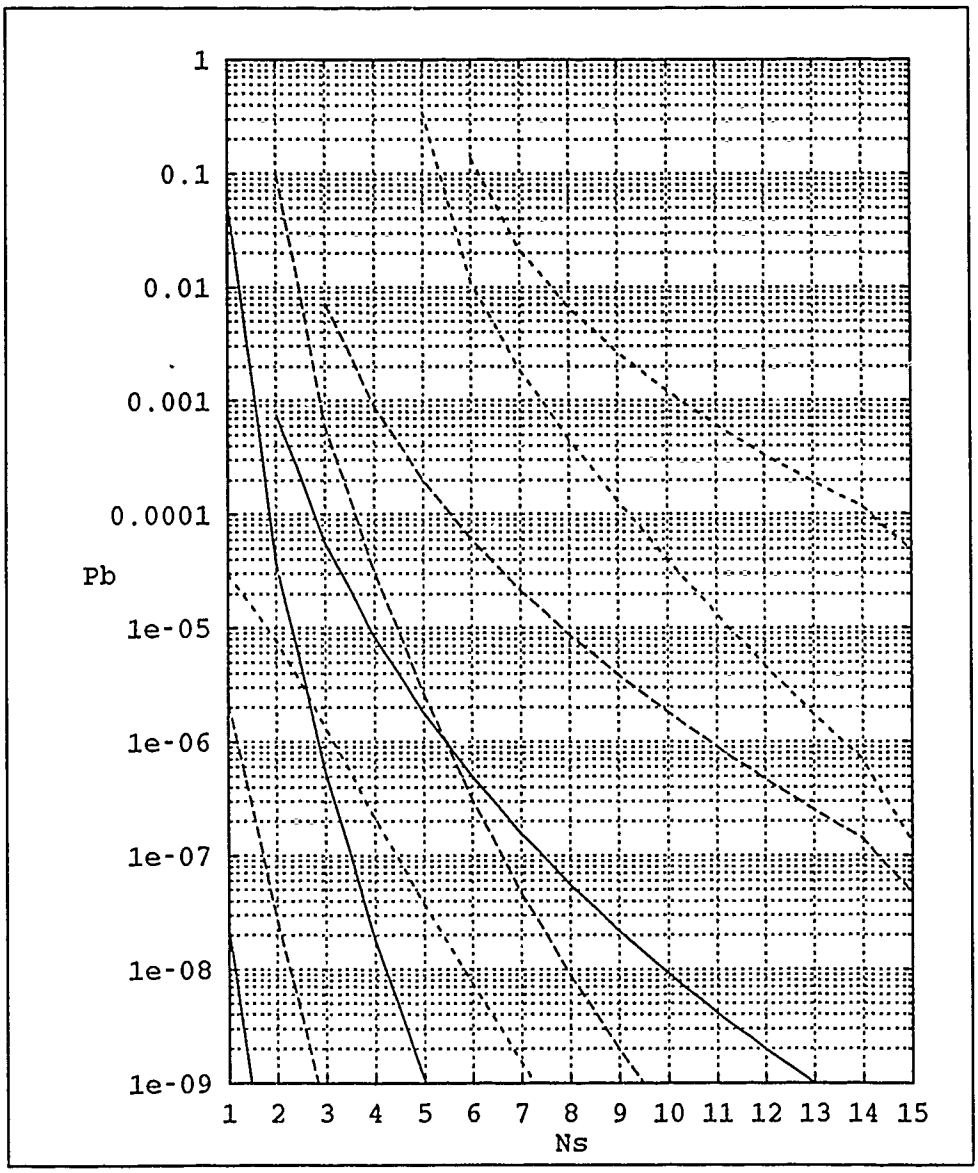


Figure 5.12. Comparing Turbulence and Background Effects for K = 7

VI. Conclusions

The objective of this thesis was to show that adaptive EDAC could combat the time varying fading caused by atmospheric turbulence. The P_b plots in Chapter V show that this is definitely acheivable. With a small number of codes with constraint lengths varying from 3 to 7 and with a laser transmitter with an intensity range of $\approx 6dB$ ($3 \leq N_s \leq 12$), P_b can be varied over a range of 5–6 orders of magnitude. The communications engineer can now assume a value of C_n^2 and a path length, calculate σ_χ^2 and pick the constraint length, N_s combination that will give the required P_b.

This thesis was the first attempt at analyzing the clear–air turbulent atmospheric optical channel for adaptive EDAC applications. There are numerous opportunities for further research. An experimental setup similar to Davidson's would be invaluable for verifying the predictions made in this thesis. It would enable analysis of the effects of varying C_n^2 and path length. Better algorithms for analyzing longer constraint lengths could be developed and other codes could then be investigated.

Appendix A. *MATRIXx Programs*

The following sections contain the programs that were used in MATRIXx to calculate the P_b for $K = 3$ through $K = 7$. The main program is the same for all cases except for calling functions that depend on the constraint length. The main program for K = 3 is shown in section A.1. The only change to the main program for a different constraint length is changing the number in the T101K3 and T100K3 functions.

A.1 Main Program

```
d =[.356998 .245867 .165898 .111127 .0742772 .0496584 .0332469 .0223035 ...
.0149948 .0101029 .00682063 .00461301 .00312477 .00211944 .00143909];
long e;
for i = 1:15,...
dpar = d(i),...
n = 1.01,...
define 't101k3',...
t101 = t101k3(n,dpar),...
n = 1.0,...
define 't100k3',...
t100 = t100k3(n,dpar),...
pb(i) = (t101 - t100)/.01,
end;
print ('pbk3s0',PB);
return
```

A.2 K3 Functions

```
//T101 = T101k3(N,DPAR)
A = [1,-N,0,0;-DPAR,1,-DPAR,0;-N*DPAR,0,1-N*DPAR,0;0,-DPAR**2,0,1];
X = [N*DPAR**2,0,0,0];
T = INV(A)*X';
```

```
T101 = T(4);
RETF

//T100 = T100k3(N,DPAR)
A = [1,-N,0,0;-DPAR,1,-DPAR,0;-N*DPAR,0,1-N*DPAR,0;0,-DPAR**2,0,1];
X = [N*DPAR**2,0,0,0];
T = INV(A)*X';
T100 = T(4);
RETF
```

A.3 K4 Functions

```
//T101 = T101k4(N,DPAR)
A = [1,-DPAR**2,-DPAR**2,0,0,0,0,0
0,1,0,-DPAR,-DPAR,0,0,0
0,0,1,0,0,-DPAR,-DPAR,0
-N*DPAR**2,0,0,1,0,0,0,0
0,-N,-N,0,1,0,0,0
0,0,0,-N*DPAR,-N*DPAR,1,0,0
0,0,0,0,0,-N*DPAR,1-N*DPAR,0
-1,0,0,0,0,0,0,1];
X = [0,0,0,N*DPAR**2,0,0,0,0];
T = INV(A)*X';
T101 = T(8);
RETF

//T100 = T100k4(N,DPAR)
A = [1,-DPAR**2,-DPAR**2,0,0,0,0,0
0,1,0,-DPAR,-DPAR,0,0,0
0,0,1,0,0,-DPAR,-DPAR,0
-N*DPAR**2,0,0,1,0,0,0,0
0,-N,-N,0,1,0,0,0
0,0,0,-N*DPAR,-N*DPAR,1,0,0
0,0,0,0,0,-N*DPAR,1-N*DPAR,0
-1,0,0,0,0,0,0,1];
```

```
X = [0,0,0,N*DPAR**2,0,0,0,0];
T = INV(A)*X';
T100= T(8);
RETF
```

A.4 K5 Functions

```
//T101 = T101k5(N,DPAR)
A = [1,-1,-DPAR**2,0,0,0,0,0,0,0,0,0,0,0,0,0
0,1,0,-DPAR**2,-1,0,0,0,0,0,0,0,0,0,0,0,0
0,0,1,0,0,-DPAR**2,-1,0,0,0,0,0,0,0,0,0,0
0,0,0,1,0,0,0,-DPAR,-DPAR,0,0,0,0,0,0,0
0,0,0,0,1,0,0,0,0,-DPAR,-DPAR,0,0,0,0,0
0,0,0,0,0,1,0,0,0,0,0,-DPAR,-DPAR,0,0,0
0,0,0,0,0,0,1,0,0,0,0,0,0,-DPAR,-DPAR,0
-N,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0
0,-N*DPAR**2,-N,0,0,0,0,0,1,0,0,0,0,0,0,0
0,0,0,-N,-N*DPAR**2,0,0,0,0,1,0,0,0,0,0,0
0,0,0,0,0,-N,-N*DPAR**2,0,0,0,1,0,0,0,0,0
0,0,0,0,0,0,0,-N*DPAR,-N*DPAR,0,0,1,0,0,0,0
0,0,0,0,0,0,0,0,0,-N*DPAR,-N*DPAR,0,1,0,0,0
0,0,0,0,0,0,0,0,0,0,0,-N*DPAR,-N*DPAR,1,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,-N*DPAR,1-N*DPAR,0
-N*DPAR**2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1];
X = [0,0,0,0,0,0,0,N*DPAR**2,0,0,0,0,0,0,0,0];
T = INV(A)*X';
T101 = T(16);
RETF

//T100 = T100k5(N,DPAR)
A = [1,-1,-DPAR**2,0,0,0,0,0,0,0,0,0,0,0,0,0
0,1,0,-DPAR**2,-1,0,0,0,0,0,0,0,0,0,0,0,0
0,0,1,0,0,-DPAR**2,-1,0,0,0,0,0,0,0,0,0,0
0,0,0,1,0,0,0,-DPAR,-DPAR,0,0,0,0,0,0,0
0,0,0,0,1,0,0,0,0,-DPAR,-DPAR,0,0,0,0,0
```

```
0,0,0,0,0,1,0,0,0,0,0,-DPAR,-DPAR,0,0,0
0,0,0,0,0,0,1,0,0,0,0,0,0,-DPAR,-DPAR,0
-N,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0
0,-N*DPAR**2,-N,0,0,0,0,0,1,0,0,0,0,0,0,0
0,0,0,-N,-N*DPAR**2,0,0,0,0,1,0,0,0,0,0,0
0,0,0,0,0,-N,-N*DPAR**2,0,0,0,1,0,0,0,0,0
0,0,0,0,0,0,0,-N*DPAR,-N*DPAR,0,0,1,0,0,0,0
0,0,0,0,0,0,0,0,0,-N*DPAR,-N*DPAR,0,1,0,0,0
0,0,0,0,0,0,0,0,0,0,0,-N*DPAR,-N*DPAR,1,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,-N*DPAR,1-N*DPAR,0
-N*DPAR**2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1];
X = [0,0,0,0,0,0,0,N*DPAR**2,0,0,0,0,0,0,0,0];
T = INV(A)*X';
T100 = T(16);
RETF
```

A.5 K6 Functions

```
//T101 = T101k6(N,DPAR)
A = [1,-DPAR**2,-1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,1,0,-1,-DPAR**2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,1,0,0,-DPAR**2,-1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,1,0,0,0,-DPAR**2,-1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,1,0,0,0,0,0,-1,-DPAR**2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,1,0,0,0,0,0,0,-DPAR**2,-1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,1,0,0,0,0,0,0,0,-1,-DPAR**2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,-DPAR,-DPAR,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,-DPAR,-DPAR,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,-DPAR,-DPAR,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,-DPAR,-DPAR,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,-DPAR,-DPAR,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,-DPAR,-DPAR,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,-DPAR,-DPAR,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,-DPAR,-DPAR,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,-DPAR,-DPAR,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,-DPAR,-DPAR,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,-DPAR,-DPAR,0
-N,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

```
0,-N,-N*DPAR**2,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,-N*DPAR**2,-N,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,-N,-N*DPAR**2,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,-N,-N*DPAR**2,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,-N*DPAR**2,-N,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,-N,-N*DPAR**2,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,-N*DPAR**2,-N,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-N*DPAR,-N*DPAR,0,0,0,0,0,0,1,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-N*DPAR,-N*DPAR,0,0,0,0,0,1,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-N*DPAR,-N*DPAR,0,0,0,0,1,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-N*DPAR,-N*DPAR,0,0, ,1,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-N*DPAR,-N*DPAR,0,0,1,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-N*DPAR,-N*DPAR,0,1,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-N*DPAR,-N*DPAR,1,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-N*DPAR,1-N*DPAR,0
-DPAR**2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1];
X = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,N*DPAR**2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
T = INV(A)*X';
T101 = T(32);
RETF

//T101 = T101k6(N,DPAR)
A = [1,-DPAR**2,-1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,1,0,-1,-DPAR**2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,1,0,0,-DPAR**2,-1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,1,0,0,0,-DPAR**2,-1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,1,0,0,0,0,-1,-DPAR**2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,1,0,0,0,0,0,-DPAR**2,-1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,1,0,0,0,0,0,0,-1,-DPAR**2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,-DPAR,-DPAR,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,-DPAR,-DPAR,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,-DPAR,-DPAR,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,-DPAR,-DPAR,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,-DPAR,-DPAR,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,-DPAR,-DPAR,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,-DPAR,-DPAR,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-DPAR,-DPAR,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-DPAR,-DPAR,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-DPAR,-DPAR,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-DPAR,-DPAR,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-DPAR,-DPAR,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-DPAR,-DPAR
```

```
0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,-DPAR,-DPAR,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,-DPAR,-DPAR,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,-DPAR,-DPAR,0
-N,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,-N,-N*DPAR**2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0 ^,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,-N*DPAR**2,-N,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,-N,-N*DPAR**2,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,-N,-N*DPAR**2,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,-N*DPAR**2,-N,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,-N,-N*DPAR**2,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,-N*DPAR**2,-N,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-N*DPAR,-N*DPAR,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-N*DPAR,-N*DPAR,0,0,0,0,0,0,1,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-N*DPAR,-N*DPAR,0,0,0,0,1,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-N*DPAR,-N*DPAR,0,0,0,1,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-N*DPAR,-N*DPAR ^,0,1,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-N*DPAR,-N*DPAR,0,1,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-N*DPAR,-N*DPAR,1,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-N*DPAR,1-N*DPAR,0
-DPAR**2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1];
X = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,N*DPAR**2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
T = INV(A)*X';
T101 = T(32);
RETF
```

A.6 K7 Functions

```
//T101 = T101k7(N,DPAR)
Z = 0*ONES(8,8);
A = [1,-DPAR**2,-DPAR**2,0,0,  ,0,0
0,1 0.-DPAR**2,-DPAR**2,0,0,0
 ,0,    .0,-1,-1,0
0,0,0,  ,  ,0,0,-1
0,0,0,0,1,0,0,0
0,0.0,0,0,1,0,0
```

```
0,0,0,0,0,0,1,0
0,0,0,0,0,0,0,1];
B = [0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
-1,0,0,0,0,0,0,0
0,-DPAR**2,-DPAR**2,0,0,0,0,0
0,0,0,-DPAR**2,-DPAR**2,0,0,0
0,0,0,0,0,-1,-1,0
0,0,0,0,0,0,0,-DPAR**2];
C = [0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
-DPAR**2,0,0,0,0,0,0,0];
D = [0,-1,-1,0,0,0,0,0
0,0,0,-1,-1,0,0,0
0,0,0,0,0,-DPAR**2,-DPAR**2,0
0,0,0,0,0,0,0,-DPAR**2
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0];
E = [0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
-DPAR**2,0,0,0,0,0,0,0
0,-1,-1,0,0,0,0,0
0,0,0,-1,-1,0,0,0
0,0,0,0,0,-DPAR**2,-DPAR**2,0
0,0,0,0,0,0,0,-DPAR**2];
```

```
F = [0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
-DPAR,0,0,0,0,0,0,0];
G = [0,-DPAR,-DPAR,0,0,0,0,0
0,0,0,-DPAR,-DPAR,0,0,0
0,0,0,0,0,-DPAR,-DPAR,0
0,0,0,0,0,0,0,-DPAR
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0];
H = [0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
-DPAR,0,0,0,0,0,0,0
0,-DPAR,-DPAR,0,0,0,0,0
0,0,0,-DPAR,-DPAR,0,0,0
0,0,0,0,0,-DPAR,-DPAR,0
0,0,0,0,0,0,0,-DPAR];
I = [0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
-DPAR,0,0,0,0,0,0,0];
J = [0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
```

```
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
-N*DPAR**2,0,0,0,0,0,0,0];
K = [0,-DPAR,-DPAR,0,0,0,0,0
0,0,0,-DPAR,-DPAR,0,0,0
0,0,0,0,0,-DPAR,-DPAR,0
0,0,0,0,0,0,0,-DPAR
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0];
L = [0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
-DPAR,0,0,0,0,0,0,0
0,-DPAR,-DPAR,0,0,0,0,0
0,0,0,-DPAR,-DPAR,0,0,0
0,0,0,0,0,-DPAR,-DPAR,0
0,0,0,0,0,0,0,0];
M = [0,-N,-N,0,0,0,0,0
0,0,0,-N,-N,0,0,0
0,0,0,0,0,-N*DPAR**2,-N*DPAR**2,0
0,0,0,0,0,0,0,-N*DPAR**2
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0];
NM = [0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
-N*DPAR**2,0,0,0,0,0,0,0
```

```
0,-N,-N,0,0,0,0,0
0,0,0,-N,-N,0,0,0
0,0,0,0,0,-N*DPAR**2,-N*DPAR**2,0
0,0,0,0,0,0,0,-N];
O = [0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
-N,0,0,0,0,0,0,0];
P = [0,-N*DPAR**2,-N*DPAR**2,0,0,0,0,0
0,0,0,-N*DPAR**2,-N*DPAR**2,0,0,0
0,0,0,0,0,-N,-N,0
0,0,0,0,0,0,0,-N
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0];
Q = [0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
-N,0,0,0,0,0,0,0
0,-N*DPAR**2,-N*DPAR**2,0,0,0,0,0
0,0,0,-N*DPAR**2,-N*DPAR**2,0,0,0
0,0,0,0,0,-N,-N,0
0,0,0,0,0,0,0,-N*DPAR];
R = [0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
```

```
0,0,0,0,0,0,0,0
-N*DPAR,0,0,0,0,0,0,0];
S = [0,-N*DPAR,-N*DPAR,0,0,0,0,0
0,0,0,-N*DPAR,-N*DPAR,0,0,0
0,0,0,0,0,-N*DPAR,-N*DPAR,0
0,0,0,0,0,0,0,-N*DPAR
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0];
T = [0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
-N*DPAR,0,0,0,0,0,0,0
0,-N*DPAR,-N*DPAR,0,0,0,0,0
0,0,0,-N*DPAR,-N*DPAR,0,0,0
0,0,0,0,0,-N*DPAR,-N*DPAR,0
0,0,0,0,0,0,0,-N*DPAR];
U = [1,0,0,0,0,0,0,0
0,1,0,0,0,0,0,0
0,0,1,0,0,0,0,0
0,0,0,1,0,0,0,0
0,0,0,0,1,0,0,0
0,0,0,0,0,1,0,0
0,0,0,0,0,0,1,0
-N*DPAR,0,0,0,0,0,0,1];
V = [0,-N*DPAR,-N*DPAR,0,0,0,0,0
0,0,0,-N*DPAR,-N*DPAR,0,0,0
0,0,0,0,0,-N*DPAR,-N*DPAR,0
0,0,0,0,0,0,0,-N*DPAR
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0];
```

```
W = [1,0,0,0,0,0,0,0
0,1,0,0,0,0,0,0
0,0,1,0,0,0,0,0
-N*DPAR,0,0,1,0,0,0,0
0,-N*DPAR,-N*DPAR,0,1,0,0,0
0,0,0,-N*DPAR,-N*DPAR,1,0,0
0,0,0,0,0,-N*DPAR,1-N*DPAR,0
0,0,0,0,0,0,0,1];
GEN = [A,B,C,Z,Z,Z,Z,Z
Z,EYE(8),D,E,F,Z,Z,Z
Z,Z,EYE(8),Z,G,H,I,Z
J,Z,Z,EYE(8),Z,Z,K,L
M,NM,O,Z,EYE(8),Z,Z,Z
Z,Z,P,Q,R,EYE(8),Z,Z
Z,Z,Z,Z,S,T,U,Z
O,Z,Z,Z,Z,Z,V,W];
X1 = [0,0,0,0,0,0,0,0];
X2 = [0,0,0,0,0,0,0,0];
X3 = [0,0,0,0,0,0,0,0];
X4 = [0,0,0,0,0,0,0,N*DPAR**2];
X5 = [0,0,0,0,0,0,0,0];
X6 = [0,0,0,0,0,0,0,0];
X7 = [0,0,0,0,0,0,0,0];
X8 = [0,0,0,0,0,0,0,0];
X = [X1,X2,X3,X4,X5,X6,X7,X8];
T = INV(GEN)*X';
T101 = T(64);
RETF

//T100 = T100k7(N,DPAR)
Z = 0*ONES(8,8);
A = [1,-DPAR**2,-DPAR**2,0,0,0,0,0
0,1,0,-DPAR**2,-DPAR**2,0,0,0
0,0,1,0,0,-1,-1,0
```

```
0,0,0,1,0,0,0,-1
0,0,0,0,1,0,0,0
0,0,0,0,0,1,0,0
0,0,0,0,0,0,1,0
0,0,0,0,0,0,0,1];
B = [0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
-1,0,0,0,0,0,0,0
0,-DPAR**2,-DPAR**2,0,0,0,0,0
0,0,0,-DPAR**2,-DPAR**2,0,0,0
0,0,0,0,0,-1,-1,0
0,0,0,0,0,0,0,-DPAR**2];
C = [0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
-DPAR**2,0,0,0,0,0,0,0];
D = [0,-1,-1,0,0,0,0,0
0,0,0,-1,-1,0,0,0
0,0,0,0,0,-DPAR**2,-DPAR**2,0
0,0,0,0,0,0,0,-DPAR**2
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0];
E = [0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
        0,0,0,0
-DPAR    0,0,0,0,0,0,0
0,-1,-1,0,0,0,0,0
```

```
0,0,0,-1,-1,0,0,0
0,0,0,0,0,-DPAR**2,-DPAR**2,0
0,0,0,0,0,0,0,-DPAR];
F = [0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
-DPAR,0,0,0,0,0,0,0];
G = [0,-DPAR,-DPAR,0,0,0,0,0
0,0,0,-DPAR,-DPAR,0,0,0
0,0,0,0,0,-DPAR,-DPAR,0
0,0,0,0,0,0,0,-DPAR
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0];
H = [0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
-DPAR,0,0,0,0,0,0,0
0,-DPAR,-DPAR,0,0,0,0,0
0,0,0,-DPAR,-DPAR,0,0,0
0,0,0,0,0,-DPAR,-DPAR,0
0,0,0,0,0,0,0,-DPAR];
I = [0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
```

```
-DPAR,0,0,0,0,0,0,0];
J = [0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
-N*DPAR**2,0,0,0,0,0,0,0];
K = [0,-DPAR,-DPAR,0,0,0,0,0
0,0,0,-DPAR,-DPAR,0,0,0
0,0,0,0,0,-DPAR,-DPAR,0
0,0,0,0,0,0,0,-DPAR
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0];
L = [0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
-DPAR,0,0,0,0,0,0,0
0,-DPAR,-DPAR,0,0,0,0,0
0,0,0,-DPAR,-DPAR,0,0,0
0,0,0,0,0,-DPAR,-DPAR,0
0,0,0,0,0,0,0,0];
M = [0,-N,-N,0,0,0,0,0
0,0,0,-N,-N,0,0,0
0,0,0,0,0,-N*DPAR**2,-N*DPAR**2,0
0,0,0,0,0,0,0,-N*DPAR**2
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0];
NM = [0,0,0,0,0,0,0,0
```

```
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
-N*DPAR**2,0,0,0,0,0,0,0
0,-N,-N,0,0,0,0,0
0,0,0,-N,-N,0,0,0
0,0,0,0,0,-N*DPAR**2,-N*DPAR**2,0
0,0,0,0,0,0,0,-N];
O = [0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
-N,0,0,0,0,0,0,0];
P = [0,-N*DPAR**2,-N*DPAR**2,0,0,0,0,0
0,0,0,-N*DPAR**2,-N*DPAR**2,0,0,0
0,0,0,0,0,-N,-N,0
0,0,0,0,0,0,0,-N
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0];
Q = [0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
-N,0,0,0,0,0,0,0
0,-N*DPAR**2,-N*DPAR**2,0,0,0,0,0
0,0,0,-N*DPAR**2,-N*DPAR**2,0,0,0
0,0,0,0,0,-N,-N,0
0,0,0,0,0,0,0,-N*DPAR];
R = [0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
```

```
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
-N*DPAR,0,0,0,0,0,0,0];
S = [0,-N*DPAR,-N*DPAR,0,0,0,0,0
0,0,0,-N*DPAR,-N*DPAR,0,0,0
0,0,0,0,0,-N*DPAR,-N*DPAR,0
0,0,0,0,0,0,0,-N*DPAR
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0];
T = [0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
-N*DPAR,0,0,0,0,0,0,0
0,-N*DPAR,-N*DPAR,0,0,0,0,0
0,0,0,-N*DPAR,-N*DPAR,0,0,0
0,0,0,0,0,-N*DPAR,-N*DPAR,0
0,0,0,0,0,0,0,-N*DPAR];
U = [1,0,0,0,0,0,0,0
0,1,0,0,0,0,0,0
0,0,1,0,0,0,0,0
0,0,0,1,0,0,0,0
0,0,0,0,1,0,0,0
0,0,0,0,0,1,0,0
0,0,0,0,0,0,1,0
-N*DPAR,0,0,0,0,0,0,1];
V = [0,-N*DPAR,-N*DPAR,0,0,0,0,0
0,0,0,-N*DPAR,-N*DPAR,0,0,0
0,0,0,0,0,-N*DPAR,-N*DPAR,0
0,0,0,0,0,0,0,-N*DPAR
0,0,0,0,0,0,0,0
```

```
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0];
W = [1,0,0,0,0,0,0,0
0,1,0,0,0,0,0,0
0,0,1.0,0,0,0,0
-N*DPAR,0,0,1,0,0,0,0
0,-N*DPAR,-N*DPAR,0,1,0,0,0
0,0,0,-N*DPAR,-N*DPAR,1,0,0
0,0,0,0,0,-N*DPAR,1-N*DPAR,0
0,0,0,0,0,0,0,1];
GEN = [A,B,C,Z,Z,Z,Z,Z
Z,EYE(8),D,E,F,Z,Z,Z
Z,Z,EYE(8),Z,G,H,I,Z
J,Z,Z,EYE(8),Z,Z,K,L
M,NM,0,Z,EYE(8),Z,Z,Z
Z,Z,P,Q,R,EYE(8),Z,Z
Z,Z,Z,Z,S,T,U,Z
0,Z,Z,Z,Z,Z,V,W];
X1 = [0.0,0,0,0,0,0,0,0];
X2 = [0,0,0,0,0,0,0,0];
X3 = [0,0,0,0,0,0,0,0];
X4 = [0,0,0.0,0,0,0,0,N*DPAR**2];
X5 = [0,0,0,0,0,0,0,0];
X6 = [0,0,0,0,0,0,0,0];
X7 = [0,0 0,0,0,0,0,0];
X8 = [0,0,0,0,0,0,0,0];
X = [X1,X2,X3,X4,X5,X6,X7,X8];
T = INV(GEN)*X';
T100 = T(64);
RETF
```

Appendix B. *C Programs*

```
#include <stdio.h>
#include <math.h>

#define pi 3.1415926535898
#define t 1.0

double xvec[] = {
-5.3874808900112,
-4.6036824495507,
-3.9447640401156,
-3.3478545673832,
-2.7888060584281,
-2.2549740020893,
-1.7385377121166,
-1.2340762153953,
-0.737[illegible]7285454,
-0.2453407083009,
0.2453407083009,
0.7374737285454,
1.23407621539[illegible],
1.7385377121166,
2.2549740020893,
2.7888060584281,
3.3478545673832,
3.9447640401156,
4.6036824495507,
5.3874808900112
};

double wvec[] = {
0.8985919614532,
0.7043329611769,
```

```
0.6222786961914,
0.5752624428525,
0.5448517423644,
0.5240803509486,
0.5096790271175,
0.4999208713363,
0.4938433852721,
0.4903215006667,
0.4909215006667,
0.4938433852721,
0.4999208713363,
0.5096790271175,
0.5240803509486,
0.5448517423644,
0.5752624428525,
0.6222786961914,
0.7043329611769,
0.8985919614532
};


double myint(ns, no, sx)
int ns;
double no, sx;
{
int i;
double ls =ns/t;
double g, sum = 0.0;

for (i = 0; i < 20; i ++)
{
g = exp(-((pow((sqrt((ls + 1)*exp(2*xvec[i]) + no)
- sqrt(no)),2.0))/(t/2.0)))*exp (-(xvec[i] * xvec[i])/(2 * sx))
        / sqrt (2 * pi *sx);
```

```
sum += wvec[i] * g;
}
return sum;
}


void make_matrixx_table(no, sx, k,filename)
double no, sx;
int  k;
char *filename;

{
int ns;
FILE *fout;

fout = fopen (filename, "w");
fprintf (fout, "d=[");
for (ns = 0;ns<14;ns++)
fprintf (fout, "%16.14e ...\n", myint (ns, no, sx));
fprintf (fout, "%16.14e];\n", myint (15, no, sx));
fprintf (fout, "long e;\nfor i = 1:15,...\ndpar
= d(i),...\nn = 1.01,...\n");
fprintf (fout, "define 't101k%d',...\nt101 =
t101k%d(n,dpar),...\n", k, k);
fprintf (fout, "n = 1.0,...\ndefine 't100k%d',...\n", k);
fprintf (fout, "t100 = t100k%d(n,dpar),...\npb(i) =
(t101 - t100)/.01,\n", k);
fprintf (fout, "end;\nprint ('pb%s',pb);\n", filename);


fclose (fout);

return;
}
```

```
void main ()
{
make_matrixx_table (0.1, .01, 3, "k3s0");
make_matrixx_table (1.0, .01, 3, "k3s0No1");
make_matrixx_table (10.0, .01, 3, "k3s0No10");
make_matrixx_table (0.1, .2, 3, "k3s2");
make_matrixx_table (1.0, .2, 3, "k3s2No1");
make_matrixx_table (10.0, .2, 3, "k3s2No10");
make_matrixx_table (0.1, .6, 3, "k3s6");
make_matrixx_table (1.0, .6, 3, "k3s6No1");
make_matrixx_table (10.0, .6, 3, "k3s6No10");
make_matrixx_table (0.1, .01, 4, "k4s0");
make_matrixx_table (1.0, .01, 4, "k4s0No1");
make_matrixx_table (10.0, .01, 4, "k4s0No10");
make_matrixx_table (0.1, .2, 4, "k4s2");
make_matrixx_table (1.0, .2, 4, "k4s2No1");
make_matrixx_table (10.0, .2, 4, "k4s2No10");
make_matrixx_table (0.1, .6, 4, "k4s6");
make_matrixx_table (1.0, .6, 4, "k4s6No1");
make_matrixx_table (10.0, .6, 4, "k4s6No10");
make_matrixx_table (0.1, .01, 5, "k5s0");
make_matrixx_table (1.0, .01, 5, "k5s0No1");
make_matrixx_table (10.0, .01, 5, "k5s0No10");
make_matrixx_table (0.1, .2, 5, "k5s2");
make_matrixx_table (1.0, .2, 5, "k5s2No1");
make_matrixx_table (10.0, .2, 5, "k5s2No10");
make_matrixx_table (0.1, .6, 5, "k5s6");
make_matrixx_table (1.0, .6, 5, "k5s6No1");
make_matrixx_table (10.0, .6, 5, "k5s6No10");
make_matrixx_table (0.1, .01, 6, "k6s0");
make_matrixx_table (1.0, .01, 6, "k6s0No1");
make_matrixx_table (10.0, .01, 6, "k6s0No10");
```

```
make_matrixx_table (0.1, .2, 6, "k6s2");
make_matrixx_table (1.0, .2, 6, "k6s2No1");
make_matrixx_table (10.0, .2, 6, "k6s2No10");
make_matrixx_table (0.1, .6, 6, "k6s6");
make_matrixx_table (1.0, .6, 6, "k6s6No1");
make_matrixx_table (10.0, .6, 6, "k6s6No10");
make_matrixx_table (0.1, .01, 7, "k7s0");
make_matrixx_table (1.0, .01, 7, "k7s0No1");
make_matrixx_table (10.0, .01, 7, "k7s0No10");
make_matrixx_table (0.1, .2, 7, "k7s2");
make_matrixx_table (1.0, .2, 7, "k7s2No1");
make_matrixx_table (10.0, .2, 7, "k7s2No10");
make_matrixx_table (0.1, .6, 7, "k7s6");
make_matrixx_table (1.0, .6, 7, "k7s6No1");
make_matrixx_table (10.0, .6, 7, "k7s6No10");
exit (0);
}
```

Appendix C. *D Parameter Data*

Table C.1. D Parameter Data

	No Turb.			Low Turb.			Medium Turb.		
	N_o			N_o			N_o		
N_s	.1	1	10	.1	1	10	.1	1	10
1	6.838 e-2	1.279 e-1	1.801 e-1	3.759 e-1	6.232 e-1	8.773 e-1	4.080 e-1	5.794 e-1	7.818 e-1
2	2.493 e-2	6.954 e-2	1.509 e-1	1.847 e-1	3.851 e-1	7.181 e-1	2.655 e-1	4.180 e-1	6.441 e-1
3	9.618 e-3	3.948 e-2	1.199 e-1	1.052 e-1	2.585 e-1	5.861 e-1	1.950 e-1	3.293 e-1	5.531 e-1
4	3.643 e-3	2.244 e-2	9.272 e-2	6.572 e-2	1.814 e-1	4.804 e-1	1.526 e-1	2.704 e-1	4.836 e-1
5	1.346 e-3	1.231 e-2	7.280 e-2	4.411 e-2	1.309 e-1	4.003 e-1	1.243 e-1	2.281 e-1	4.304 e-1
6	4.871 e-4	6.496 e-3	5.793 e-2	3.115 e-2	9.757 e-2	3.395 e-1	1.040 e-1	1.967 e-1	3.891 e-1
7	1.734 e-4	3.305 e-3	4.636 e-2	2.274 e-2	7.511 e-2	2.912 e-1	8.882 e-2	1.726 e-1	3.555 e-1
8	6.097 e-5	1.630 e-3	3.694 e-2	1.696 e-2	5.948 e-2	2.510 e-1	7.692 e-2	1.534 e-1	3.268 e-1
9	2.120 e-5	7.837 e-4	2.910 e-2	1.287 e-2	4.810 e-2	2.166 e-1	6.744 e-2	1.377 e-1	3.015 e-1
10	7.309 e-6	3.682 e-4	2.258 e-2	9.911 e-3	3.944 e-2	1.872 e-1	5.976 e-2	1.245 e-1	2.792 e-1
11	2.499 e-6	1.696 e-4	1.723 e-2	7.740 e-3	3.263 e-2	1.622 e-1	5.344 e-2	1.131 e-1	2.596 e-1
12	8.494 e-7	7.679 e-5	1.293 e-2	6.132 e-3	2.714 e-2	1.412 e-1	4.816 e-2	1.034 e-1	2.423 e-1
13	2.869 e-7	3.422 e-5	9.554 e-3	4.928 e-3	2.267 e-2	1.236 e-1	4.370 e-2	9.490 e-2	2.272 e-1
14	9.641 e-8	1.504 e-5	6.948 e-3	4.016 e-3	1.900 e-2	1.091 e-1	3.987 e-2	8.750 e-2	2.139 e-1
15	1.074 e-8	2.800 e-6	3.518 e-3	2.769 e-3	1.352 e-2	8.702 e-2	3.365 e-2	7.531 e-2	1.918 e-1

Appendix D. P_b *data*

The following five tables document the P_b results. These tables include all calculated values, even if $N_s < N_{s_{min}}$. Each table contains nine data files. Each table contains data files for one constraint length, three different levels of turbulence, and three different levels of background light. The data from these files is what was used to generate the plots in Chapter V.

Table D.1. P_b Data for K = 3

P_b For $K = 3$			
N_s	No Turb., $N_o = .1$	No Turb., $N_o = 1$	No Turb., $N_o = 10$
1	$1.761182406657526\ 10^{-6}$	$4.904566565602125\ 10^{-5}$	$3.396456792650220\ 10^{-4}$
2	$1.017354924451575\ 10^{-8}$	$1.921734348660993\ 10^{-6}$	$1.224958709911291\ 10^{-4}$
3	$8.397319078211453\ 10^{-11}$	$1.048272598181580\ 10^{-7}$	$3.331418923555024\ 10^{-5}$
4	$6.472600105900685\ 10^{-13}$	$5.969356340095553\ 10^{-9}$	$8.678203923716110\ 10^{-6}$
5	$4.433856655295366\ 10^{-15}$	$2.910710890111120\ 10^{-10}$	$2.438359965463750\ 10^{-6}$
6	$2.747571891027707\ 10^{-17}$	$1.173029370202268\ 10^{-11}$	$7.469991627962507\ 10^{-7}$
7	$1.571882969699214\ 10^{-19}$	$3.971810008318049\ 10^{-13}$	$2.379987415479274\ 10^{-7}$
8	$8.431511677911093\ 10^{-22}$	$1.157133576249314\ 10^{-14}$	$7.469081871613898\ 10^{-8}$
9	$4.288591615981567\ 10^{-24}$	$2.962219528746474\ 10^{-16}$	$2.223642918456624\ 10^{-8}$
10	$2.085955427953614\ 10^{-26}$	$6.778258264562654\ 10^{-18}$	$6.159539365369669\ 10^{-9}$
11	$9.764961200283160\ 10^{-29}$	$1.405706140023401\ 10^{-19}$	$1.575958127837387\ 10^{-9}$
12	$4.421784567919114\ 10^{-31}$	$2.672098384729986\ 10^{-21}$	$3.720772305832319\ 10^{-10}$
13	$1.944613366786925\ 10^{-33}$	$4.699436067252185\ 10^{-23}$	$8.121053675096788\ 10^{-11}$
14	$8.332955515699094\ 10^{-36}$	$7.706607805107916\ 10^{-25}$	$1.643361047373747\ 10^{-11}$
15	$1.430307877339532\ 10^{-40}$	$1.723468515013948\ 10^{-28}$	$5.430471929799078\ 10^{-13}$
N_s	Low Turb., $N_o = .1$	Low Turb., $N_o = 1$	Low Turb., $N_o = 10$
1	$8.120055014739710\ 10^{-2}$	$8.487659687208049\ 10^{-1}$	$1.677902410456666\ 10^{0}$
2	$3.931892282434991\ 10^{-4}$	$1.082286983237514\ 10^{-1}$	$7.696636284990427\ 10^{-1}$
3	$1.700529670121647\ 10^{-5}$	$3.314796936897471\ 10^{-3}$	$1.106801331367360\ 10^{0}$
4	$1.433177452290168\ 10^{-6}$	$3.537424615151867\ 10^{-4}$	$6.342054048982953\ 10^{1}$
5	$1.845648168273705\ 10^{-7}$	$5.576841310259403\ 10^{-5}$	$1.801994950094393\ 10^{-1}$
6	$3.142249121934885\ 10^{-8}$	$1.137747267107984\ 10^{-5}$	$2.848829344308801\ 10^{-2}$
7	$6.384467711497623\ 10^{-9}$	$2.870491703555186\ 10^{-6}$	$7.875677543651548\ 10^{-3}$
8	$1.457612075765014\ 10^{-9}$	$8.557950975777767\ 10^{-7}$	$2.709052136780571\ 10^{-3}$
9	$3.631626773978695\ 10^{-10}$	$2.873272713027236\ 10^{-7}$	$1.038846812090549\ 10^{-3}$
10	$9.763100487750492\ 10^{-11}$	$1.042681984068186\ 10^{-7}$	$4.265466560153942\ 10^{-4}$
11	$2.823742398542454\ 10^{-11}$	$3.972916049777352\ 10^{-8}$	$1.850424482966563\ 10^{-4}$
12	$8.784030800346874\ 10^{-12}$	$1.562163872009560\ 10^{-8}$	$8.457867363303756\ 10^{-5}$
13	$2.937118961913594\ 10^{-12}$	$6.285513800467057\ 10^{-9}$	$4.074988505470772\ 10^{-5}$
14	$1.053208729634933\ 10^{-12}$	$2.581357298837393\ 10^{-9}$	$2.069959431916303\ 10^{-5}$
15	$1.639785267277624\ 10^{-13}$	$4.655390295906908\ 10^{-10}$	$6.209172645633017\ 10^{-6}$
N_s	Med Turb., $N_o = .1$	Med Turb., $N_o = 1$	Med Turb., $N_o = 10$
1	$2.379629443127217\ 10^{-1}$	$1.195674781507047\ 10^{0}$	$9.271490168948904\ 10^{-1}$
2	$3.992748811609001\ 10^{-3}$	$3.546305080024785\ 10^{-1}$	$7.867306842671201\ 10^{-1}$
3	$5.441960393269649\ 10^{-4}$	$2.154722825671706\ 10^{-2}$	$1.795235665863260\ 10^{0}$
4	$1.306331513774349\ 10^{-4}$	$4.556196067477102\ 10^{-3}$	$2.259715112525575\ 10^{2}$
5	$4.196162484604749\ 10^{-5}$	$1.441293220643628\ 10^{-3}$	$5.964349212153219\ 10^{-1}$
6	$1.606196044084013\ 10^{-5}$	$5.726198895502450\ 10^{-4}$	$1.234850134312394\ 10^{-1}$
7	$6.916003649504374\ 10^{-6}$	$2.642157249132219\ 10^{-4}$	$4.454524327495591\ 10^{-2}$
8	$3.250888820356039\ 10^{-6}$	$1.347803248022064\ 10^{-4}$	$2.015718277251066\ 10^{-2}$
9	$1.639654355303673\ 10^{-6}$	$7.362405206986415\ 10^{-5}$	$1.032264965290606\ 10^{-2}$
10	$8.771171516624087\ 10^{-7}$	$4.224100177599823\ 10^{-5}$	$5.745679712175315\ 10^{-3}$
11	$4.931641502445706\ 10^{-7}$	$2.517739630895961\ 10^{-5}$	$3.412336555122953\ 10^{-3}$
12	$2.892425784560671\ 10^{-7}$	$1.549766482689044\ 10^{-5}$	$2.139584527041704\ 10^{-3}$
13	$1.758168348759075\ 10^{-7}$	$9.818900461017002\ 10^{-6}$	$1.405331785242145\ 10^{-3}$
14	$1.101574339911617\ 10^{-7}$	$6.389580652569720\ 10^{-6}$	$9.603416451709336\ 10^{-4}$
15	$4.654814508057136\ 10^{-8}$	$2.911160472784485\ 10^{-6}$	$4.925816172545805\ 10^{-4}$

Table D.2. P_b Data for K = 4

P_b For $K = 3$			
N_s	No Turb., $N_o = .1$	No Turb., $N_o = 1$	No Turb., $N_o = 10$
1	$1.761182406657526\ 10^{-6}$	$4.904566565602125\ 10^{-5}$	$3.396456792650220\ 10^{-4}$
2	$1.017354924451575\ 10^{-8}$	$1.921734348660993\ 10^{-6}$	$1.224958709911291\ 10^{-4}$
3	$8.397319078211453\ 10^{-11}$	$1.048272598181580\ 10^{-7}$	$3.331418923555024\ 10^{-5}$
4	$6.472600105900685\ 10^{-13}$	$5.969356340095553\ 10^{-9}$	$8.678203923716110\ 10^{-6}$
5	$4.433856655295366\ 10^{-15}$	$2.910710890111120\ 10^{-10}$	$2.438359965463750\ 10^{-6}$
6	$2.747571891027707\ 10^{-17}$	$1.173029370202268\ 10^{-11}$	$7.469991627962507\ 10^{-7}$
7	$1.571882969699214\ 10^{-19}$	$3.971810008318049\ 10^{-13}$	$2.379987415479274\ 10^{-7}$
8	$8.431511677911093\ 10^{-22}$	$1.157133576249314\ 10^{-14}$	$7.469081871613898\ 10^{-8}$
9	$4.288591615981567\ 10^{-24}$	$2.962219528746474\ 10^{-16}$	$2.223642918456624\ 10^{-8}$
10	$2.085955427953614\ 10^{-26}$	$6.778258264562654\ 10^{-18}$	$6.159539365369669\ 10^{-9}$
11	$9.764961200283160\ 10^{-29}$	$1.405706140023401\ 10^{-19}$	$1.575958127837387\ 10^{-9}$
12	$4.421784567919114\ 10^{-31}$	$2.672098384729986\ 10^{-21}$	$3.720772305832319\ 10^{-10}$
13	$1.944613366786925\ 10^{-33}$	$4.699436067252185\ 10^{-23}$	$8.121053675096788\ 10^{-11}$
14	$8.332955515699094\ 10^{-36}$	$7.706607805107916\ 10^{-25}$	$1.643361047373747\ 10^{-11}$
15	$1.430307877339532\ 10^{-40}$	$1.723468515013948\ 10^{-28}$	$5.430471929799078\ 10^{-13}$
N_s	Low Turb., $N_o = .1$	Low Turb., $N_o = 1$	Low Turb., $N_o = 10$
1	$8.120055014739710\ 10^{-2}$	$3.487659687208049\ 10^{-1}$	$1.677902410456666\ 10^{0}$
2	$3.931892282434991\ 10^{-4}$	$1.082286983237514\ 10^{-1}$	$7.696636284990427\ 10^{-1}$
3	$1.700529670121647\ 10^{-5}$	$3.314796936897471\ 10^{-3}$	$1.106801331367360\ 10^{0}$
4	$1.433177452290168\ 10^{-6}$	$3.537424615151867\ 10^{-4}$	$6.342054048982953\ 10^{1}$
5	$1.845649168273705\ 10^{-7}$	$5.576841310259403\ 10^{-5}$	$1.801994950094393\ 10^{-1}$
6	$3.142249121934885\ 10^{-8}$	$1.137747267107984\ 10^{-5}$	$2.848829344308801\ 10^{-2}$
7	$6.384467711497623\ 10^{-9}$	$2.870491703555186\ 10^{-6}$	$7.875677543651548\ 10^{-3}$
8	$1.457612975765014\ 10^{-9}$	$8.557950975777767\ 10^{-7}$	$2.709052136780571\ 10^{-3}$
9	$3.631626773978695\ 10^{-10}$	$2.873272713027236\ 10^{-7}$	$1.038846812090549\ 10^{-3}$
10	$9.763100487750492\ 10^{-11}$	$1.042681984068186\ 10^{-7}$	$4.265466560153942\ 10^{-4}$
11	$2.823742398542454\ 10^{-11}$	$3.972916049777352\ 10^{-8}$	$1.850424482966563\ 10^{-4}$
12	$8.784030800346874\ 10^{-12}$	$1.562163872009560\ 10^{-8}$	$8.457867363303756\ 10^{-5}$
13	$2.937118961913594\ 10^{-12}$	$6.285513800467057\ 10^{-9}$	$4.074988505470772\ 10^{-5}$
14	$1.053208729634933\ 10^{-12}$	$2.581357298837393\ 10^{-9}$	$2.069959431916303\ 10^{-5}$
15	$1.639785267277624\ 10^{-13}$	$4.655390295906908\ 10^{-10}$	$6.209172645633017\ 10^{-6}$
N_s	Med Turb., $N_o = .1$	Med Turb., $N_o = 1$	Med Turb., $N_o = 10$
1	$2.379629443127217\ 10^{-1}$	$1.195674781507047\ 10^{0}$	$9.271490168948904\ 10^{-1}$
2	$3.992748811609001\ 10^{-3}$	$3.546305080024785\ 10^{-1}$	$7.867306842671201\ 10^{-1}$
3	$5.441960393269649\ 10^{-4}$	$2.154722825671706\ 10^{-2}$	$1.795235665863260\ 10^{0}$
4	$1.306331513774349\ 10^{-4}$	$4.556196067477102\ 10^{-3}$	$2.259715112525575\ 10^{2}$
5	$4.196162484604749\ 10^{-5}$	$1.441293220643628\ 10^{-3}$	$5.964349212153219\ 10^{-1}$
6	$1.606196044084013\ 10^{-5}$	$5.726198895502450\ 10^{-4}$	$1.234850134312394\ 10^{-1}$
7	$6.916003649504374\ 10^{-6}$	$2.642157249132219\ 10^{-4}$	$4.454524327495591\ 10^{-2}$
8	$3.250888820356039\ 10^{-6}$	$1.347803248022064\ 10^{-4}$	$2.015718277251066\ 10^{-2}$
9	$1.639654355303673\ 10^{-6}$	$7.362405206986415\ 10^{-5}$	$1.032264965290606\ 10^{-2}$
10	$8.771171516624087\ 10^{-7}$	$4.224100177599823\ 10^{-5}$	$5.745679712175315\ 10^{-3}$
11	$4.931641502445706\ 10^{-7}$	$2.517739630895961\ 10^{-5}$	$3.412336555122953\ 10^{-3}$
12	$2.892425784560671\ 10^{-7}$	$1.549766482689044\ 10^{-5}$	$2.139584527041704\ 10^{-3}$
13	$1.758168348759075\ 10^{-7}$	$9.818900461017002\ 10^{-6}$	$1.405331785242145\ 10^{-3}$
14	$1.101574339911617\ 10^{-7}$	$6.389580652569720\ 10^{-6}$	$9.603416451709336\ 10^{-4}$
15	$4.654814508057136\ 10^{-8}$	$2.911160472784485\ 10^{-6}$	$4.925816172545805\ 10^{-4}$

Table D.3. P_b Data for K = 5

P_b For $K = 3$			
N_s	No Turb., $N_o = .1$	No Turb., $N_o = 1$	No Turb., $N_o = 10$
1	$3.588845176773725\ 10^{-7}$	$1.801184503768433\ 10^{-5}$	$1.680465130267246\ 10^{-4}$
2	$7.669038761387091\ 10^{-10}$	$3.980163380726879\ 10^{-7}$	$5.207015185079078\ 10^{-5}$
3	$2.446419436436495\ 10^{-12}$	$1.247034326894963\ 10^{-8}$	$1.147238600864936\ 10^{-5}$
4	$7.146243354404256\ 10^{-15}$	$4.051058721145679\ 10^{-10}$	$2.366512627595794\ 10^{-6}$
5	$1.808640864230535\ 10^{-17}$	$1.085829575287451\ 10^{-11}$	$5.278314017989475\ 10^{-7}$
6	$4.056102188233781\ 10^{-20}$	$2.308813460411382\ 10^{-13}$	$1.295504847913651\ 10^{-7}$
7	$8.262857754515191\ 10^{-23}$	$3.977720907063443\ 10^{-15}$	$3.317402180471925\ 10^{-8}$
8	$1.557876481989643\ 10^{-25}$	$5.717712803349629\ 10^{-17}$	$8.318987758782499\ 10^{-9}$
9	$2.755690890547344\ 10^{-28}$	$7.034889943650598\ 10^{-19}$	$1.954607725392788\ 10^{-9}$
10	$4.619803437705312\ 10^{-31}$	$7.563678831881658\ 10^{-21}$	$4.206064372755250\ 10^{-10}$
11	$7.400280816438396\ 10^{-34}$	$7.226038558594962\ 10^{-23}$	$8.219595965743579\ 10^{-11}$
12	$1.137425173407888\ 10^{-36}$	$6.218221367756666\ 10^{-25}$	$1.457409248334728\ 10^{-11}$
13	$1.716024625870967\ 10^{-39}$	$4.874214330825844\ 10^{-27}$	$2.350232400630777\ 10^{-12}$
14	$1.131821909637099\ 10^{-42}$	$3.513042449737500\ 10^{-29}$	$3.459435483874404\ 10^{-13}$
15	$1.336613602433585\ 10^{-46}$	$1.462674764641170\ 10^{-33}$	$5.789118843527627\ 10^{-15}$
N_s	Low Turb., $N_o = .1$	Low Turb., $N_o = 1$	Low Turb., $N_o = 10$
1	$1.112505747432867\ 10^{-1}$	$3.645008647716608\ 10^{-2}$	$-1.297001939504244\ 10^{-1}$
2	$1.986239552879131\ 10^{-4}$	$1.735913277627548\ 10^{-1}$	$-1.343486519955056\ 10^{-2}$
3	$5.220252950621978\ 10^{-6}$	$2.218733031621265\ 10^{-3}$	$6.048894486972989\ 10^{-2}$
4	$2.810136834317699\ 10^{-7}$	$1.760390946552966\ 10^{-4}$	$4.232164042545797\ 10^{-1}$
5	$2.449585741767166\ 10^{-8}$	$2.091636336584438\ 10^{-5}$	$4.313034274341834\ 10^{-1}$
6	$2.955676813040052\ 10^{-9}$	$3.255309287777114\ 10^{-6}$	$2.719520709096677\ 10^{-2}$
7	$4.390605358812497\ 10^{-10}$	$6.403854195633279\ 10^{-7}$	$5.929377005524828\ 10^{-3}$
8	$7.485702453305598\ 10^{-11}$	$1.522872728180790\ 10^{-7}$	$1.766789109960466\ 10^{-3}$
9	$1.415655429742433\ 10^{-11}$	$4.152646824989154\ 10^{-8}$	$5.988339013774150\ 10^{-4}$
10	$2.930919049102203\ 10^{-12}$	$1.239097453498596\ 10^{-8}$	$2.179569747221685\ 10^{-4}$
11	$6.621426663135924\ 10^{-13}$	$3.912452551063778\ 10^{-9}$	$8.376239766728474\ 10^{-5}$
12	$1.632017401396818\ 10^{-13}$	$1.281217910795444\ 10^{-9}$	$3.392294739359754\ 10^{-5}$
13	$4.385698529831749\ 10^{-14}$	$4.309259054064914\ 10^{-10}$	$1.451365962390224\ 10^{-5}$
14	$1.281530040667071\ 10^{-14}$	$1.484465853269795\ 10^{-10}$	$6.573972458374902\ 10^{-6}$
15	$1.376239930903162\ 10^{-15}$	$1.906482908117359\ 10^{-11}$	$1.594551718750755\ 10^{-6}$
N_s	Med Turb., $N_o = .1$	Med Turb., $N_o = 1$	Med Turb., $N_o = 10$
1	$7.885245478859597\ 10^{-1}$	$6.612648734597576\ 10^{-2}$	$-5.279573991508191\ 10^{-2}$
2	$2.738044105650970\ 10^{-3}$	$2.446917658423788\ 10^{0}$	$2.518265434996725\ 10^{-2}$
3	$2.876358773413516\ 10^{-4}$	$1.930808588654748\ 10^{-2}$	$9.278919042043851\ 10^{-2}$
4	$5.608346337297015\ 10^{-5}$	$3.179356740378413\ 10^{-3}$	$3.788577707523860\ 10^{-1}$
5	$1.501815469693323\ 10^{-5}$	$8.669529860511201\ 10^{-4}$	$7.888675059683534\ 10^{1}$
6	$4.881993564268271\ 10^{-6}$	$3.047802679370277\ 10^{-4}$	$2.158718824316941\ 10^{-1}$
7	$1.810875825513764\ 10^{-6}$	$1.260725060765081\ 10^{-4}$	$4.815927308105591\ 10^{-2}$
8	$7.420511338777283\ 10^{-7}$	$5.814292437280116\ 10^{-5}$	$1.781422226651555\ 10^{-2}$
9	$3.296925255066699\ 10^{-7}$	$2.888256612107963\ 10^{-5}$	$8.097102819865998\ 10^{-3}$
10	$1.568091169499945\ 10^{-7}$	$1.513479966289268\ 10^{-5}$	$4.136945599605393\ 10^{-3}$
11	$7.903780365572642\ 10^{-8}$	$8.268861750717344\ 10^{-6}$	$2.292611438951678\ 10^{-3}$
12	$4.185664232452487\ 10^{-8}$	$4.681223928538208\ 10^{-6}$	$1.353884242526594\ 10^{-3}$
13	$2.311712113573773\ 10^{-8}$	$2.737054316171640\ 10^{-6}$	$8.425694484550083\ 10^{-4}$
14	$1.323110279446483\ 10^{-8}$	$1.649374424006569\ 10^{-6}$	$5.478898389982230\ 10^{-4}$
15	$4.728023384668811\ 10^{-9}$	$6.511444169874516\ 10^{-7}$	$2.567994539756703\ 10^{-4}$

Table D.4. P_b Data for K = 6

P_b For $K = 3$			
N_s	No Turb., $N_o = .1$	No Turb., $N_o = 1$	No Turb., $N_o = 10$
1	$2.382127307176335\ 10^{-5}$	$2.231174284297292\ 10^{-6}$	$2.971355535087142\ 10^{-5}$
2	$1.885098299156822\ 10^{-11}$	$2.685947826201847\ 10^{-8}$	$7.638074861784263\ 10^{-6}$
3	$2.338866274725782\ 10^{-14}$	$4.822446476683164\ 10^{-10}$	$1.322674451991810\ 10^{-6}$
4	$2.597894551151169\ 10^{-17}$	$8.971983476719887\ 10^{-12}$	$2.122301074413344\ 10^{-7}$
5	$2.432694545137668\ 10^{-20}$	$1.327454414624348\ 10^{-13}$	$3.726342876442348\ 10^{-8}$
6	$1.975465431803489\ 10^{-23}$	$1.493787718307394\ 10^{-15}$	$7.304223399876416\ 10^{-9}$
7	$1.432550075619665\ 10^{-26}$	$1.311940624735994\ 10^{-17}$	$1.502435890064750\ 10^{-9}$
8	$9.427133212499651\ 10^{-30}$	$9.314232260044080\ 10^{-20}$	$3.013036943800496\ 10^{-10}$
9	$4.390171200035802\ 10^{-33}$	$5.510861355375329\ 10^{-22}$	$5.595893982260899\ 10^{-11}$
10	$-6.829907668649090\ 10^{-36}$	$2.785015473498183\ 10^{-24}$	$9.372634491312656\ 10^{-12}$
11	$-6.102907131704052\ 10^{-38}$	$1.226271236584166\ 10^{-26}$	$1.401822796484459\ 10^{-12}$
12	$-3.190332457977668\ 10^{-41}$	$4.773036242179096\ 10^{-29}$	$1.870238706283164\ 10^{-13}$
13	$-1.600907898305692\ 10^{-44}$	$1.470028954599524\ 10^{-31}$	$2.232070504786458\ 10^{-14}$
14	$-7.746733446803838\ 10^{-48}$	$8.757901655161567\ 10^{-34}$	$2.393246054009976\ 10^{-15}$
15	$-1.650446056129215\ 10^{-54}$	$-1.351955786795744\ 10^{-37}$	$2.032193798223647\ 10^{-17}$
N_s	Low Turb., $N_o = .1$	Low Turb., $N_o = 1$	Low Turb., $N_o = 10$
1	$5.743664664500233\ 10^{-2}$	$2.659093328550022\ 10^{-2}$	$1.270522175501332\ 10^{-2}$
2	$3.606910679826663\ 10^{-5}$	$9.973576791457144\ 10^{-2}$	$1.498327504435987\ 10^{-2}$
3	$5.309820419209465\ 10^{-7}$	$5.909969593869129\ 10^{-4}$	$3.397367088680131\ 10^{-2}$
4	$1.793677628647456\ 10^{-8}$	$3.135830529243702\ 10^{-5}$	$1.320314549901232\ 10^{-1}$
5	$1.056452915164334\ 10^{-9}$	$2.653360716068365\ 10^{-6}$	$3.332709571871491\ 10^{-1}$
6	$9.050546551583628\ 10^{-11}$	$3.071416476355582\ 10^{-7}$	$1.077408259820985\ 10^{-2}$
7	$9.852770990751123\ 10^{-12}$	$4.662676338210941\ 10^{-8}$	$1.842621245179735\ 10^{-3}$
8	$1.257210509321701\ 10^{-12}$	$8.811736495408450\ 10^{-9}$	$4.540243831457460\ 10^{-4}$
9	$1.807980926719250\ 10^{-13}$	$1.950008395977639\ 10^{-9}$	$1.296463118689234\ 10^{-4}$
10	$2.886941955968257\ 10^{-14}$	$4.786807601157403\ 10^{-10}$	$4.017051017898418\ 10^{-5}$
11	$5.100324452423429\ 10^{-15}$	$1.253868662464033\ 10^{-10}$	$1.325418499208964\ 10^{-5}$
12	$9.969320132346569\ 10^{-16}$	$3.424219821258997\ 10^{-11}$	$4.647741271967976\ 10^{-6}$
13	$2.154608109469527\ 10^{-16}$	$9.640711118410.41\ 10^{-12}$	$1.737108236745757\ 10^{-6}$
14	$5.133228135621588\ 10^{-17}$	$2.789452413443463\ 10^{-12}$	$6.936649046796719\ 10^{-7}$
15	$3.805204741211403\ 10^{-18}$	$2.557150877844377\ 10^{-13}$	$1.342835958718455\ 10^{-7}$
N_s	Med Turb., $N_o = .1$	Med Turb.. $N_o = 1$	Med Turb., $N_o = 10$
1	$8.241289142822711\ 10^{-1}$	$3.576630460481109\ 10^{-2}$	$1.046862481187821\ 10^{-2}$
2	$7.538455212590924\ 10^{-4}$	$8.798089953412937\ 10^{0}$	$2.342114954357391\ 10^{-2}$
3	$5.540986245500433\ 10^{-5}$	$7.231698424961953\ 10^{-3}$	$4.427587837375485\ 10^{-2}$
4	$8.324699564395067\ 10^{-6}$	$8.961280180452228\ 10^{-4}$	$1.218374264515466\ 10^{-1}$
5	$1.807291092696138\ 10^{-6}$	$1.990548074458836\ 10^{-4}$	$6.013387701980970\ 10^{0}$
6	$4.913101530064253\ 10^{-7}$	$5.925685001413348\ 10^{-5}$	$1.316470239336894\ 10^{-1}$
7	$1.556237210469711\ 10^{-7}$	$2.129316359944198\ 10^{-5}$	$2.105792179801268\ 10^{-2}$
8	$5.531498954225660\ 10^{-8}$	$8.680138704528980\ 10^{-6}$	$6.586058016557492\ 10^{-3}$
9	$2.158870038080265\ 10^{-8}$	$3.857084205125528\ 10^{-6}$	$2.642124493447580\ 10^{-3}$
10	$9.116049391667563\ 10^{-9}$	$1.823572134293640\ 10^{-6}$	$1.215162366372452\ 10^{-3}$
11	$4.116355274622969\ 10^{-9}$	$9.049358413713859\ 10^{-7}$	$6.138316514995245\ 10^{-4}$
12	$1.968021342102399\ 10^{-9}$	$4.679675146589373\ 10^{-7}$	$3.335956822324857\ 10^{-4}$
13	$9.877141033860166\ 10^{-10}$	$2.512122523783557\ 10^{-7}$	$1.925825601366948\ 10^{-4}$
14	$5.165878504579067\ 10^{-10}$	$1.396505996971246\ 10^{-7}$	$1.169515771512415\ 10^{-4}$
15	$1.562538235892088\ 10^{-10}$	$4.753641391946674\ 10^{-8}$	$4.858330376914143\ 10^{-5}$

Table D.5. P_b Data for K = 7

P_b For $K = 3$			
N_s	No Turb., $N_o = .1$	No Turb., $N_o = 1$	No Turb., $N_o = 10$
1	$2.382127344268520\ 10^{-6}$	$2.231177437699840\ 10^{-6}$	$2.971409116624136\ 10^{-5}$
2	$1.885098299347114\ 10^{-11}$	$2.685947873331373\ 10^{-8}$	$7.638111220814427\ 10^{-6}$
3	$2.338866263838570\ 10^{-14}$	$4.822446478409413\ 10^{-10}$	$2.226675566213365\ 10^{-6}$
4	$2.597894751999962\ 10^{-17}$	$8.971983475273210\ 10^{-12}$	$2.122301365414041\ 10^{-7}$
5	$2.432695279028801\ 10^{-20}$	$1.327454412475624\ 10^{-13}$	$3.726342966960358\ 10^{-8}$
6	$1.975484127927358\ 10^{-23}$	$1.493787728667977\ 10^{-15}$	$7.304223434812596\ 10^{-9}$
7	$1.433290331900348\ 10^{-26}$	$1.311940683553941\ 10^{-17}$	$1.502435891493983\ 10^{-9}$
8	$9.499167544345708\ 10^{-30}$	$9.314237940559646\ 10^{-20}$	$3.013036944567887\ 10^{-10}$
9	$5.843646755618153\ 10^{-33}$	$5.510852355491087\ 10^{-22}$	$5.595893982819536\ 10^{-11}$
10	$3.376597671210300\ 10^{-36}$	$2.784751489577215\ 10^{-24}$	$9.372634491624909\ 10^{-12}$
11	$1.849243461916648\ 10^{-39}$	$1.225746588087130\ 10^{-26}$	$1.401822795604751\ 10^{-12}$
12	$9.667029172946557\ 10^{-43}$	$4.775306103028195\ 10^{-29}$	$1.870238707520325\ 10^{-13}$
13	$4.850911495913263\ 10^{-46}$	$1.668395817368469\ 10^{-31}$	$2.232070513031348\ 10^{-14}$
14	$2.347337778672049\ 10^{-49}$	$5.285132649300303\ 10^{-34}$	$2.393246080687990\ 10^{-15}$
15	$5.001016612943432\ 10^{-56}$	$4.096565130858280\ 10^{-39}$	$2.032193774244956\ 10^{-17}$
N_s	Low Turb., $N_o = .1$	Low Turb., $N_o = 1$	Low Turb., $N_o = 10$
1	$5.824923614211278\ 10^{-2}$	$2.417387030075810\ 10^{-2}$	$9.316316159344029\ 10^{-3}$
2	$3.606989252995464\ 10^{-5}$	$1.018044453478983\ 10^{-1}$	$1.282253105658882\ 10^{-2}$
3	$5.309822229150763\ 10^{-7}$	$5.911832549244163\ 10^{-4}$	$3.139590114074053\ 10^{-2}$
4	$1.793677649692370\ 10^{-8}$	$3.135890128443548\ 10^{-5}$	$1.242493588168456\ 10^{-1}$
5	$1.056452915912844\ 10^{-9}$	$2.653365166996777\ 10^{-6}$	$3.484696869598895\ 10^{-1}$
6	$9.050546551417740\ 10^{-11}$	$3.071417084369847\ 10^{-7}$	$1.081707350598432\ 10^{-2}$
7	$9.852770990838970\ 10^{-12}$	$4.662676479826909\ 10^{-8}$	$1.844257967413326\ 10^{-3}$
8	$1.257210508153758\ 10^{-12}$	$8.811736546414640\ 10^{-9}$	$4.541363213076704\ 10^{-4}$
9	$1.807980925588390\ 10^{-13}$	$1.950008398448117\ 10^{-9}$	$1.296560493342801\ 10^{-4}$
10	$2.886941951223575\ 10^{-14}$	$4.786807602464710\ 10^{-10}$	$4.017148200598792\ 10^{-5}$
11	$5.100324448461079\ 10^{-15}$	$1.253868662576982\ 10^{-10}$	$1.325429344884891\ 10^{-5}$
12	$9.969320140927946\ 10^{-16}$	$3.424219821591307\ 10^{-11}$	$4.647754832646934\ 10^{-6}$
13	$2.154608172258122\ 10^{-16}$	$9.640711115665297\ 10^{-12}$	$1.737110153353517\ 10^{-6}$
14	$5.133227699673478\ 10^{-17}$	$2.789452413741822\ 10^{-12}$	$6.936652129091910\ 10^{-7}$
15	$3.805204902308587\ 10^{-18}$	$2.557150879581083\ 10^{-13}$	$1.342836075536997\ 10^{-7}$
N_s	Med Turb., $N_o = .1$	Med Turb., $N_o = 1$	Med Turb., $N_o = 10$
1	$8.900673734152466\ 10^{-1}$	$3.313992078286046\ 10^{-2}$	$8.478781911559763\ 10^{-3}$
2	$1.431603395345\ 10^{-4}$	$1.202220882925985\ 10^{1}$	$2.105949429850110\ 10^{-2}$
3	$5.541169474610012\ 10^{-5}$	$7.252539989458179\ 10^{-3}$	$4.137056498324937\ 10^{-2}$
4	$8.324742695118008\ 10^{-6}$	$8.965427753358741\ 10^{-4}$	$1.147267622247223\ 10^{-1}$
5	$1.807293166444064\ 10^{-6}$	$1.990773337652379\ 10^{-4}$	$4.703434192633561\ 10^{0}$
6	$4.913103080562563\ 10^{-7}$	$5.925894130770415\ 10^{-5}$	$1.349380162431932\ 10^{-1}$
7	$1.556237367233361\ 10^{-7}$	$2.129344086599585\ 10^{-5}$	$2.120017548430369\ 10^{-2}$
8	$5.531499153427659\ 10^{-8}$	$8.680185565183570\ 10^{-6}$	$6.603621975059663\ 10^{-3}$
9	$2.158870068548048\ 10^{-8}$	$3.857093567532652\ 10^{-6}$	$2.645351729816206\ 10^{-3}$
10	$9.116049446076360\ 10^{-9}$	$1.823574245373506\ 10^{-6}$	$1.215904752701084\ 10^{-3}$
11	$4.116355285741972\ 10^{-9}$	$9.049363647592486\ 10^{-7}$	$6.140320737379708\ 10^{-4}$
12	$1.968021344677371\ 10^{-9}$	$4.679676553764520\ 10^{-7}$	$3.336572619354721\ 10^{-4}$
13	$9.877141040231267\ 10^{-10}$	$2.512122931061629\ 10^{-7}$	$1.926036783228483\ 10^{-4}$
14	$5.165876506375393\ 10^{-10}$	$1.396506123287158\ 10^{-7}$	$1.169595335994243\ 10^{-4}$
15	$1.562538236124325\ 10^{-10}$	$4.753641539174235\ 10^{-8}$	$4.858471781701328\ 10^{-5}$

Bibliography

1. Air Force Wright Aeronautical Laboratories. *Turbulence Modulation of Laser Beam Propagation from Airborne Platforms.* AFWAL-TR-85-1017. Alexandria VA: Defense Technical Information Center, 1985 (AD-B0976236).
2. Beland, Robert R. et al. "Comparison of Horizontal Scintillation Measurements and Models, *Proceedings of the SPIE, 926*: 44–51 1988.
3. Beland, Robert. Personal Interview. Atmospheric Optics Branch, Air Force Geophysics Lab, Hanscom AFB July 25,1990. 1990.
4. Brown, James H. and Neil J. Grossbard. "Short Term Prediction of Optical Turbulence Parameters," *Proceedings of the SPIE, 926*: 278–285, 1988.
5. Chan, V.W.S., "Coding for the Atmospheric Optical Channel," *IEEE Transactions on Communications 30*: 269–274 (January 1982).
6. Cheng, Jeng-Shiang. *Performance Analysis for Randomly Dispersive Optical Channels.* PhD dissertation. Sever Institute of Technology, Washington University, St. Louis MO, August 1983.
7. Chernov, Lev A. *Wave Propagation in a Random Media.* New York: McGraw Hill Book Company, 1960.
8. Clark, George C. Jr. and J. Bibb Cain *Error Correction Coding for Digital Communications.* New York: Plenum Press, 1981.
9. Clifford, S. F. "The Classical Theory of Wave Propagation in a Turbulent Medium," in *Laser Beam Propagation in the Atmosphere* (J. W. Strohbehn, editor), Springer–Verlag, Heidelberg (1978).
10. Clifford, S. F. *Wave Propagation in a Turbulent Medium* PhD dissertation, Dartmouth College, Hanover, N. H. (1969).
11. Corrsin, S. "On the Spectrum of Isotropic Temperature Fluctuations in an Isotropic Turbulence." *Journal of Applied Physics, 22*:469 (1951).
12. Davidson, Frederick and Yutai T. Koh. "Interleaved Convolutional Coding for the Turbulent Atmospheric Optical Channel," *IEEE Transactions on Communications, 36*: 993–1003 (September 1988).
13. Feldman, Robert J. *Airborne Laser Communications Scintillation Measurements.* MS thesis, AFIT/GEO/ENG/87J-2. School of Engineering, Air Force Institute of Technology (AU), Wright Patterson AFB OH, June 1987 (AD-B113375).
14. Feldman, Robert J. and Steven K. Rogers. "Extended Communication Path Length Scintillation Measurements and Model: a Comparison of Results," *Proceedings of the SPIE, 1115*: 28–37 (1989).

15. Forestieri, Enrico et al. "Performance of Convolutional Codes in a Direct Detection Optical PPM Channel," *IEEE Transactions on Communications, 37*: 1303–1317 (December 1989).

16. Gagliardi, Robert M. and Sherman Karp. *Optical Communications.* New York: John Wiley and Sons, 1976.

17. Good, R. E. et al. "Atmospheric Models of Optical Turbulence," *Proceedings of the SPIE, 928*: 165–186 (1988).

18. Goodman, Joesph W. *Statistical Optics.* New York: John Wiley and Sons, 1985.

19. Greenwood, Darryl P. Videotape of short course, *Optical Propagation*, Massachusetts Institute of Technology, 1982.

20. Holt, John C. *The Effects of Atmospheric Turbulence on the Laser Airborne Communications Experiment, (HAVE LACE).* MS thesis, AFIT/GE/ENG/87D–26. School of Engineering, Air Force Institute of Technology (AU), Wright Patterson AFB OH, December 1987 (AD–B117441).

21. Holt, John C. et al. Executive Summary for HAVE LACE test program.

22. Hoversten, E. V. et al. "Communication theory for the Turbulent Atmosphere," *Proceedings of the IEEE, 58*: 1626- 1650 (October 1988).

23. Kanavos, Jay N. *The Effects of Atmospheric Turbulence on an Air-to-air Optical Communications Link.* MS thesis, AFIT/GE/ENG/84D–38. School of Engineering, Air Force Institute of Technology (AU), Wright Patterson AFB OH, December 1984 (AD–A151840).

24. Kerr J.R. et al. "Atmospheric Optical Communications Systems," *Proceedings of the IEEE, 58* : 1691- 1709 (XXX 19xx)

25. Koh, Yutai T. and Frederic Davidson. "Interleaved Concatenated Coding for the Turbulent Atmospheric Direct Detection Optical Communication Channel," *IEEE Transactions on Communications, 37*: 648–651 (June 1989).

26. Kolmogorov, A. N. "The Local Structure of Turbulence in Incompressible Viscous Fluid for Very Large Reynolds' Numbers," *Doklady Akad. Nauk SSSR 30*:301 (1941).

27. Kravtsov, Y. A. *Izv. VUZ, Radiofiz. 13*: 281 English translation: *Radiophys. Quantum Electronic 13*: 217 (1970).

28. Lin, Shu and Daniel Costello. *Error Control Coding: Fundamentals and Applications.* Englewood Cliffs NJ: Prentice Hall Inc., 1983.

29. Massey, J. L. "Capacity Cutoff Rate and Coding for a Direct Detection Optical Channel," *IEEE Transactions on Communications, 29*:260-269 (1967).

30. Massey, J. L. and Daniel J. Costello "Nonsystematic Convolutional Codes for Sequential Decoding in Space Applications," *IEEE Transactions on Communications Technology, COM-19*: 806-813 (1971).

31. Mathematica, Wolfram Research Inc., Steve Wolfram, 1988.

32. MathStation Versions 1.1 and 1.2, MathSoft Inc. Cambridge MA, 1990

33. MATRIXx Version 7.0, Integrated Systems Inc. Santa Clara, CA, 1988.

34. Murphy, Edmund A. et al. " Scintillation and Turbulence Measurements: Comparisons Over a Horizontal Path," *Proceedings of the SPIE, 926*: 352-359 (1988).

35. Omura, J. K. "On the Viterbi Decoding Algorithm" (coorespondence), *IEEE Transactions on Information Theory, IT-15*: 177-179 (1969).

36. Personick, Stewart D. "Fundamental Limits in Optical Communications," *Proceedings of the IEEE, 69*: 262-266 (February 1981).

37. Pratt, W. K. *Laser Communications Systems*, New York: John Wiley and Sons Inc., 1969.

38. Saleh, A. A. M. "An Investigation of Laser Wave Depolarization Due to Atmospheric Transmission." *Journal of Quantum Electronics, 3*: 540 (1967).

39. Shannon, Claude E. *The Mathematical Theory of Communication.* Urbana, IL: University of Illinois Press, 1949.

40. Sklar, Bernard *Digital Communications: Fundamentals and Applications.* Englewood Cliffs NJ: Prentice Hall, 1988.

41. Strohbehn, J. W. editor *Laser Beam Propagation in the Atmosphere.* New York: Springer-Verlag, 1978.

42. Strohbehn, J. W. "Line of Sight Propagation Through the Turbulent Atmosphere" *Proceedings of the IEEE, 56*:1304 (1968).

43. Strohbehn, J. W. and Clifford, S. F., "Polarization and Angle-of-Arrival Fluctuations for a Plane Wave Propagated Through a Turbulent Medium", *Transactions on Antennas and Propagation,15*: 416 (1967).

44. Tatarski, V. I. *Wave Propagation in a Turbulent Medium*, New York: McGraw Hill (1976).

45. Tatarski V. I. *Izv. VUZ, Radiofiz. 10*: 1762 (1967); English translation: *Radiophys. Quantum Electronic 10*: 987 (1967).

46. Turner, Vernon. "Atmospheric Scintillation Measurements with a Transmissometer," *Proceedings of the SPIE, 926*: 39-43 (1988).

47. Viterbi, Andrew. J. and Jim K. Ómura *Principles of Digital Communication and Coding.* New York: Mcgraw Hill, 1979.

48. Viterbi, Andrew J. "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithim," *IEEE Transactions on Information Theory, IT- 13*: 260-269 (1967).

49. Viterbi, Andrew J. "Convolutional Codes and their Performance in Communication Systems," *IEEE Transactions on Communications Technology, COM-19*: 751-772 (1971).

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1990	Master's Thesis

4. TITLE AND SUBTITLE
USING ERROR DETECTION AND CORRECTION CODING FOR A TURBULENT ATMOSPHERIC OPTICAL COMMUNICATIONS LINK

5. FUNDING NUMBERS

6. AUTHOR(S)
Mark A. Cloutier
Captain, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology
WPAFB OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GEO/ENG/90D-3

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Robert Feldman
WRDC/AAAI
WPAFB, OH 45433

10. SPONSORING MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Probability of bit error (P_b) performance for a turbulent, atmospheric, optical communications link with convolutional error detection and correction coding was investigated. The codes investigated were all rate 1/2 codes with constraint lengths ranging from 3 to 7. It was assumed that the coded data stream was sufficiently interleaved so that the turbulent channel could be considered memoryless. A theoretical bound on P_b, given by Viterbi for a memoryless channel, was used to plot curves of P_b versus the signal intensity of the laser transmitter. These curves were plotted for three different levels of atmospheric turbulence and three different levels of background light.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Atmospheric Turbulence, Error Detection and Correction Coding, Optical Communications, Convolutional Coding	98
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL